# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**HARDWARE IN THE LOOP IMPLEMENTATION OF ADAPTIVE VISION BASED GUIDANCE LAW FOR GROUND TARGET TRACKING**

by

Soh Mun Lok Bernard

December 2008


| | |
|---|---|
| Thesis Advisor: | Vladimir N. Dobrokhodov |
| Co-Advisor: | Kevin D. Jones |

THIS PAGE INTENTIONALLY LEFT BLANK

| REPORT DOCUMENTATION PAGE | *Form Approved OMB No. 0704-0188* |
|---|---|



| **1. AGENCY USE ONLY *(Leave blank)*** | **2. REPORT DATE** December 2008 | **3. REPORT TYPE AND DATES COVERED** Master's Thesis |
|---|---|---|

| | |
|---|---|
| **4. TITLE AND SUBTITLE** Hardware In The Loop Implementation of Adaptive Vision Based Guidance Law for Ground Target Tracking | **5. FUNDING NUMBERS** |
| **6. AUTHOR(S)** Soh Mun Lok Bernard | |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)** Naval Postgraduate School Monterey, CA 93943-5000 | **8. PERFORMING ORGANIZATION REPORT NUMBER** |
| **9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)** N/A | **10. SPONSORING/MONITORING AGENCY REPORT NUMBER** |

**11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

| **12a.** Approved for public release; distribution is unlimited | **12b.** A |
|---|---|

**13. ABSTRACT (maximum 200 words)**

An adaptive guidance law of a Vision Based Target Tracking (VBTT) system was previously developed and implemented onboard a Small Unmanned Aerial Vehicle (SUAV) in order to track a ground target moving with a constant velocity. This work extends previous results by considering scenarios where the variation of target velocity, in both magnitude and direction, is used to excite the feedback control law for further robustness analysis. This provides essential insight on the sensitivity of the performance criteria indicated by the range holding capability, navigation error and the convergence speed of the guidance law.

In addition, this thesis addresses the robustness of the SUAV guidance law to the generalized time delay in feedback due to, for example, data processing or communications lag. This thesis also extends the previously obtained results by introducing a multi-criteria optimization technique. The results obtained are first based on the numerical simulations implemented in SIMULINK and then in high fidelity HIL simulation environment with Piccolo Plus AP in the control loop.

Initial steps in developing Vision Based HIL environment incorporating TASE gimbal, Piccolo Plus AP, Pan-Tilt unit and image processing software are presented. The work also includes motivation for the development, an overview of the existing technologies, and initial implementation of low-level driving mechanism (drivers) for the realistic representation of the real-world environment.

| | |
|---|---|
| **14. SUBJECT TERMS** Unmanned Aerial Vehicle, UAV, Vision Based Target Tracking, Autonomous Guidance, xPC Target, PC-104, Hardware in the Loop Simulation | **15. NUMBER OF PAGES** 109 |
| | **16. PRICE CODE** |

| **17. SECURITY CLASSIFICATION OF REPORT** | **18. SECURITY CLASSIFICATION OF THIS PAGE** | **19. SECURITY CLASSIFICATION OF ABSTRACT** | **20. LIMITATION OF ABSTRACT** |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UU |

NSN 7540-01-280-5500 Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

# HARDWARE IN THE LOOP IMPLEMENTATION OF ADAPTIVE VISION BASED GUIDANCE LAW FOR GROUND TARGET TRACKING

Soh Mun Lok Bernard
Captain, Singapore Armed Forces (Army)
B.Eng (MPE), Nanyang Technological University, 2003

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN MECHANICAL ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL**
**December 2008**

Author: Soh Mun Lok Bernard

Approved by: Vladimir N. Dobrokhodov, PhD
Thesis Advisor

Kevin D. Jones, PhD
Co-Advisor

Knox T. Millsaps, PhD
Chairman, Department of Mechanical and Astronautical Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

An adaptive guidance law of a Vision Based Target Tracking (VBTT) system was previously developed and implemented onboard a Small Unmanned Aerial Vehicle (SUAV) in order to track a ground target moving with a constant velocity. This work extends previous results by considering scenarios where the variation of target velocity, in both magnitude and direction, is used to excite the feedback control law for further robustness analysis. This provides essential insight on the sensitivity of the performance criteria indicated by the range holding capability, navigation error and the convergence speed of the guidance law.

In addition, this thesis addresses the robustness of the SUAV guidance law to the generalized time delay in feedback due to, for example, data processing or communications lag. This thesis also extends the previously obtained results by introducing a multi-criteria optimization technique. The results obtained are first based on the numerical simulations implemented in SIMULINK and then in high fidelity HIL simulation environment with Piccolo Plus AP in the control loop.

Initial steps in developing Vision Based HIL environment incorporating TASE gimbal, Piccolo Plus AP, Pan-Tilt unit and image processing software are presented. The work also includes motivation for the development, an overview of the existing technologies, and initial implementation of low-level driving mechanism (drivers) for the realistic representation of the real-world environment.

THIS PAGE INTENTIONALLY LEFT BLANK

## TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF FIGURES

# LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# ACKNOWLEDGMENTS

The author would like to thank Professor Vladimir Dobrokhodov for all his valuable help and patience in explaining the difficult concepts contained in this thesis. He has provided invaluable guidance with the laboratory hardware in the loop setup and the subsequent integration and implementation of the adaptive vision based target tracking algorithm. The author is grateful to Professor Vladimir for imparting his research experience and techniques and making the entire project a wonderful experiential learning experience.

The author would also like to thank Mr Kam Khim Yee and Mr Jaya Kumar for their companionship during the seemingly endless hours of SIMULINK and hardware in the loop simulations in the laboratory.

Finally, the author would like to thank his wife for her encouragement and understanding in the completion of this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

## A. BACKGROUND

Since the turn of the millennium, many military and security forces around the world have been modernizing their equipment and operating platforms in preparation for future network centric warfare concepts. Coupled with technological advances in communications, solid-state sensors and power supplies, this modernization resulted in an unprecedented use of Unmanned Aerial Vehicles (UAVs) in modern battlefield. Used predominantly in the areas of intelligence, reconnaissance, surveillance, battle damage assessments and target tracking missions, UAVs are able to operate in all weather conditions and environments and offer better endurance and persistence over human deployment.

For the US military in particular, the wars in Afghanistan and Iraq further saw the usage of UAVs across the U.S. military network as a whole leap to 258,000 hours in 2007. This number is expected to surge to 300,000 hours in 2008.[1] Faced with the ever growing demand for actionable intelligence in an evolving battlefield, human operators of UAV will inevitably need to multi-task and work harder in order to control autonomous platforms, process and deliver timely information.

To aid in this information management process, a fully autonomous unmanned aerial platform will be indispensable to enhance the effectiveness of UAV operations and to allow more spare capacity to the human operator. Sophisticated target detection and a vision based tracking algorithm have been in active development in order to address this need. The key task at hand then

[1] Armed Forces International, “US Military's UAV Missions Increasing” http://www.armedforces-int.com/news/2008/01/02/us-militarys-uav-missions-increasing.asp, [Accessed Nov 08].

becomes the integration of the maturing hardware technologies and the advanced guidance and control algorithms to truly achieve the goal of unmanned autonomous surveillance.

## B. OVERVIEW OF THE RESEARCH PROJECT

This thesis is part of an ongoing effort within the Unmanned Systems Research Group at the Naval Postgraduate School (NPS) to build a Vision Based Target Tracking (VBTT) simulator system in a laboratory environment.

Figure 1. NPS Small UAV, Sig Rascal

This VBTT simulator system will imitate actual UAV flight dynamics of a small UAV seen above in Figure 1. This provides an ideal research and development and training tool that can be used in the lab. The key benefit of this setup is that it will greatly facilitate the development, integration and testing of new vision-based target tracking software and *guidance, navigation and control* (GNC) algorithms in a safe laboratory environment. It will considerably reduce the likelihood of a failure by detecting bugs and deficiencies before the tactical UAV aircraft is put at risk in an actual flight test.

The VBTT simulator system is an elaborate setup that comprises both hardware and software components. The principal functions of the various components are briefly described as follows:

1. **Piccolo Autopilot (AP) System** – The AP system together with the 6-DOF flight simulator application provides simulated flight dynamics inputs to the GNC algorithm that is used to drive the motion of the UAV.

2. **Six Degree of Freedom (DOF) Flight Model** – Is used to provide simulated flight dynamics in a test environment.

3. **Pan-Tilt Unit (PTU)** – Is used to simulate the aircraft dynamics for an onboard gimbal in flight condition and is driven by the commands from the AP.

4. **TASE Gimbal** – The TASE gimbal is a small inertially stabilized gimbal with an onboard GPS and IMU for standalone operations. It is the first image capturing device in the VBTT setup and is mounted on the PTU unit. The imagery captured by the TASE gimbal is subsequently transmitted to the ground where it is processed by the automated motion tracking software.

5. **Motion Tracking Software** – The automated motion tracking software is developed by PerceptiVU Inc. Once the user selects the target of interest, the software is used to track the target automatically without any further user intervention.

6. **GNC Algorithm** – The algorithm [1] solves two principal tasks. First, it autonomously navigates the UAV around the target while keeping the target in the camera frame. Second, it controls the gimbaled camera installed onboard so that when the vision-based tracker is engaged, the target is kept in the center of the camera frame. The target motion estimation error is minimized when the target moves parallel to the camera image plane.

This control strategy used in the development of the guidance law is depicted in Figure 2 [1]. It illustrates a 2-D shape of an SUAV orbit above the target, while the UAV is autonomously guided to accomplish the task of target tracking. The guidance algorithm controls the orientation of the UAV ground speed vector $V_g$ to make it continuously perpendicular to the line of sight (LOS), where the LOS is the line connecting the UAV and the target.

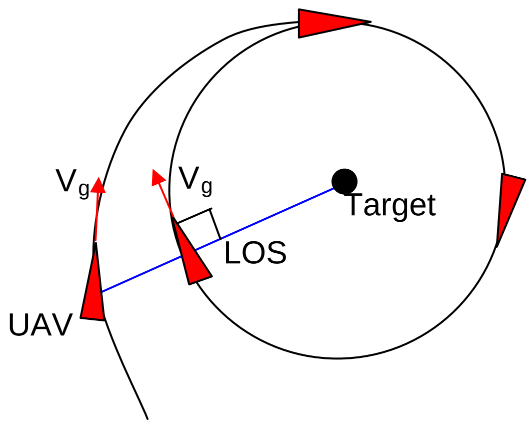



Figure 2. Conceptual Representation of VBTT Guidance

## C. SCOPE AND OBJECTIVES

The objective of this thesis is to further investigate the robustness of the developed guidance law to the variations of target escape maneuver and to develop and integrate several components required for a Hardware-In-The-Loop (HIL) simulation. The scope of work is as follows:

- Investigate performance of the guidance control law subject to a target escape maneuver.
- Develop and integrate new adaptive control law into the existing 6-DOF UAV flight dynamics model and implement it on a PC/104 form factor computer.
- Develop all the necessary software drivers and integrate the corresponding hardware components required for HIL simulation.
- Perform real-time HIL simulation of the adaptive control law and prove its feasibility for further flight implementation.

# II. DEVELOPMENT OF CONTROL LAW

## A. COORDINATE SYSTEMS

Multiple coordinate frames are often used to define the kinematics motion and behavior of an object in complex dynamic systems. Hence the translational and rotational kinematics states (i.e. the positions, velocities, and accelerations) of the objects can be fully described in any frame as long as the transformation relationships between the coordinate frames are known. The following coordinate systems were used in the development of the control law in this thesis [2] – [6].

### 1. Navigation Inertial Coordinate Frame (I –Frame)

This local level frame assumes a flat earth model in the vicinity of the reference navigation point: it has $X_n$ – $Y_n$ axes in a plane tangent (known as Local Tangent Plane, LTP) to the reference point origin on the earth ellipsoid surface, and the $Z_n$ axis perpendicular to that ellipsoid surface. It is assumed that the $X_n$ axis points north, the $Y_n$ axis east, and the $Z_n$ axis down. This is a north-east-down (NED) coordinate system and complies with the right-hand rule as illustrated in Figure 3.

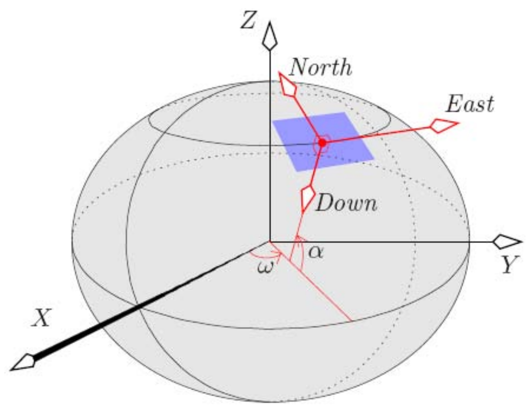



Figure 3. NED Coordinate System

### 2. SUAV Body Coordinate Frame (B – Frame)

A convenient coordinate system for developing the equations of motion of the SUAV is a right-hand orthogonal system with its origin centered at the aircraft center of gravity. Conventionally, the x-axis points forward along the longitudinal axis of the aircraft, the y-axis points outwards towards the right wing, and the z-axis is in the downward direction as seen in Figure 4.

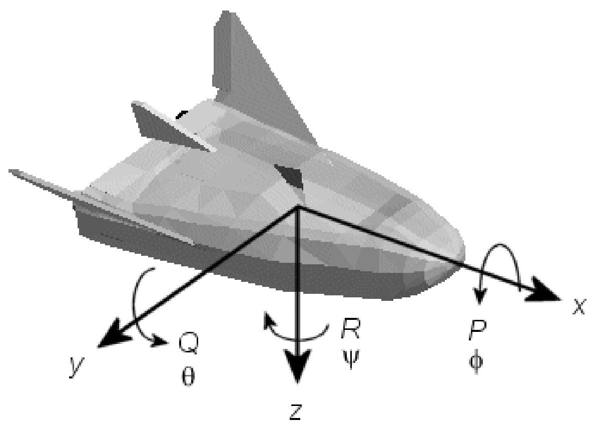



Figure 4. UAV Body Coordinate Frame (B – Frame)

### 3. Gimbal Platform Coordinate Frame (G – Frame)

The gimbal coordinate frame is a right-hand orthogonal coordinate system with the origin being the location of the camera mount. The x-axis of the gimbal frame points forward along the longitudinal axis of the gimbal platform, the y-axis points outward toward the right-hand side, and the z-axis points downward from the gimbal platform.

### 4. Camera Coordinate Frame (C – Frame)

The camera coordinate frame is a right-hand orthogonal coordinate system with the origin located at the focal point of the camera. The x-axis points forward along the longitudinal axis of the camera, the y-axis points outwards toward the right-hand side, and the z-axis points downward from the origin.

### 5. Image Plane Coordinate Frame (P –Frame)

The image plane reference frame is the coordinate system used to describe the location of the target in the image plane. It is a 2-dimensional coordinate system with the u-axis aligned with the y-axis of the camera coordinate frame, and the v-axis aligned with the negative z-axis of the camera coordinate frame.

## B. EULER ANGLES

Euler angles are the classical way of representing rotations in 3-dimensional Euclidean space. The triplet of Euler angles $(\phi,\theta,\psi)$ relates two orthogonal coordinate systems having a common origin and the coordinate frames can be transformed to the other through a series of rotations defined by the Euler angles. Conventionally when the Euler angles $(\phi,\theta,\psi)$ are used to describe the orientation of the aircraft body in relation to the inertia frame, these Euler angle are known as roll, pitch and yaw respectively and are shown in Figure 4 [2], [3], [4].

## C. RELATIONSHIPS BETWEEN COORDINATE FRAMES

A coordinate transformation is a conversion from one coordinate system to another, to describe the same object. A rotation is a type of transformation from one system of coordinates to another system of coordinates such that the distance between any two points remains invariant under the transformation. A rigid body position in space can be represented by a [3 x 1] vector, and its orientation to its own current coordinate frame or a transformed coordinate frame can be uniquely described by a [3 x 3] rotation matrix at any instant in time [2], [3].

## 1. Rotation Matrices

The rotation matrices for a single 2-dimensional rotation about each individual axis are given below. The angle of rotation is the Euler angle that corresponds to each individual axis.

$$R_X(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \tag{1}$$

$$R_Y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{2}$$

$$R_Z(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

The complete rotation or transformation of the coordinate system is the sequential combination of the 2-dimensional rotations about each axis. The sequence or order of the rotation is important to properly define the orientation of the body and to preserve the orthogonality of orientation (right-hand system or left- hand system) of the transformed axes.

## 2. Coordinates Transformation

The coordinate transformation or rotation from the inertia frame to the camera frame can be obtained via sequential coordinate rotation from one frame to the other in the correct logical order as shown below:

$$ {}_I^C R = {}_G^C R \, {}_B^G R \, {}_I^B R \tag{4}$$

where ${}_I^B R$ is the coordinate rotation from inertia frame to body frame

${}_B^G R$ is the coordinate rotation from body frame to gimbal frame

${}_G^C R$ is the coordinate rotation from gimbal frame to camera frame

### *a. Inertial Frame to Body Frame Transformation*

The coordinate transformation from the inertial frame to the body frame is simply the product of the three individual rotation matrices:

$$ {}_{I}^{B}R = R_x\left({}^{I}\phi_B\right) R_y\left({}^{I}\theta_B\right) R_z\left({}^{I}\psi_B\right) \tag{5}$$

### *b. Body Frame to Gimbal Platform Frame Transformation*

The coordinate transformation from the body frame to the gimbal platform frame only involves rotation through two angles because the gimbal platform is a two-axis coordinate system. There is no rotation along the x-axis (roll rotation) and hence the roll angle is taken to be zero. The gimbal platform frame will be rotated along the y-axis (pitch rotation) and the z-axis (yaw rotation) as shown in the following rotation:

$$ {}_{B}^{G}R = R_y\left({}^{B}\theta_G\right) R_z\left({}^{B}\psi_G\right) \tag{6}$$

### *c. Gimbal Platform Frame to Camera Frame Transformation*

The coordinate transformation between the gimbal platform frame and the camera frame allows for any misalignment angles that exist between the mounting of the camera platform with the gimbal platform. In most likelihood, some or all of the rotation angles will be zero because the axes will be directly aligned:

$$ {}_{G}^{C}R = R_x\left({}^{G}\phi_C\right) R_y\left({}^{G}\theta_C\right) R_z\left({}^{G}\psi_C\right) \tag{7}$$

### d. Camera Frame to Image Plane Frame Transformation

The coordinate transformation between the camera frame and the image plane frame is not a rotational but a positional transformation. The position [u, v] of an object in the image plane frame from a position in the camera frame is given as below and is also seen in Figure 5.

$$\begin{bmatrix} P_u \\ P_v \end{bmatrix} = \frac{f}{c_x} \begin{bmatrix} c_y \\ -c_z \end{bmatrix} \tag{8}$$

It is of interest to note that, unlike the previous rotational transformations, the camera to image plane transformation is irreversible. This is due to the transformation of a three-dimensional coordinate system to a two-dimensional coordinate system.

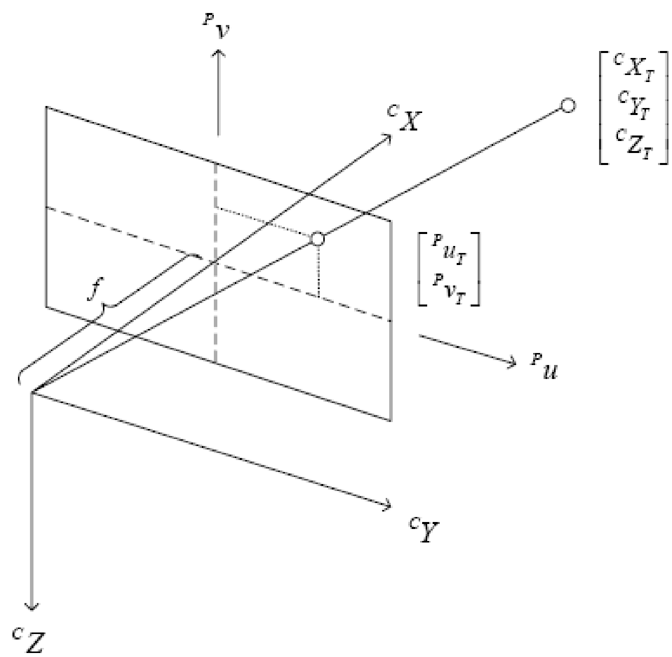



Figure 5. Image Plane Reference Frame

## 3. Angular Velocities Transformations

### *a. Body Frame Angular Velocities with Respect to Inertia Frame*

The angular velocity vector $\omega$, in the body fixed coordinates system of the SUAV, has components [p, q, r] in the x, y and z direction respectively. The relationships with the Euler angle rates $\left[\dot{\phi}_B,\, \dot{\theta}_B,\, \dot{\psi}_B\right]$ for which are stated as follow:

$$
{}^{B}\omega_{BI} = R_x\left({}^{I}\phi_B\right)R_y\left({}^{I}\theta_B\right)R_z\left({}^{I}\psi_B\right)\begin{bmatrix}0\\0\\\dot{\psi}_B\end{bmatrix} + R_x\left({}^{I}\phi_B\right)R_y\left({}^{I}\theta_B\right)\begin{bmatrix}0\\\dot{\theta}_B\\0\end{bmatrix} + R_x\left({}^{I}\phi_B\right)\begin{bmatrix}\dot{\phi}_B\\0\\0\end{bmatrix}
$$

$$
{}^{B}\omega_{BI} = \begin{bmatrix}\omega_x\\\omega_y\\\omega_z\end{bmatrix} = \begin{bmatrix}p_B\\q_B\\r_B\end{bmatrix} = \begin{bmatrix}\dot{\psi}_x + \dot{\theta}_x + \dot{\phi}_x\\\dot{\psi}_y + \dot{\theta}_y + \dot{\phi}_y\\\dot{\psi}_z + \dot{\theta}_z + \dot{\phi}_z\end{bmatrix} = \begin{bmatrix}\dot{\phi}_B - \dot{\psi}_B \sin {}^{I}\theta_B\\\dot{\psi}_B \cos {}^{I}\theta_B \sin {}^{I}\phi_B + \dot{\theta}_B \cos {}^{I}\phi_B\\\dot{\psi}_B \cos {}^{I}\theta_B \sin {}^{I}\phi_B - \dot{\theta}_B \cos {}^{I}\phi_B\end{bmatrix} \qquad (9)
$$

Expressing $\left[\dot{\phi}_B,\, \dot{\theta}_B,\, \dot{\psi}_B\right]$ in terms of [$p_B$, $q_B$, $r_B$] below, we observe the singularity problem in Equation (10) when ${}^{I}\theta_B = \pm 90^{o}$:

$$
\begin{bmatrix}\dot{\psi}_B\\\dot{\theta}_B\\\dot{\phi}_B\end{bmatrix} = \begin{bmatrix}\dfrac{1}{\cos\left({}^{I}\theta_B\right)}\left[q_B \sin\left({}^{I}\phi_B\right) + r_B \cos\left({}^{I}\phi_B\right)\right]\\q_B \cos\left({}^{I}\phi_B\right) - r_B \sin\left({}^{I}\phi_B\right)\\p_B + \tan\left({}^{I}\theta_B\right)\left[q_B \sin\left({}^{I}\phi_B\right) + r_B \cos\left({}^{I}\phi_B\right)\right]\end{bmatrix} \qquad (10)
$$

### *b. Gimbal Frame Angular Velocities with Respect to Inertia Frame*

The angular velocities of the gimbal frame with respect to the inertia frame are shown below. Equation (10) relates the angular velocity expressed in the gimbal platform frame while Equation (11) describes the transformation of the same angular rate in the inertia frame:

$$ {}^{G}\omega_{GI} \square \begin{bmatrix} p_G \\ q_G \\ r_G \end{bmatrix} = R_y\left({}^{I}\theta_G\right) R_z\left({}^{I}\psi_G\right) \begin{bmatrix} p_B \\ q_B \\ r_B \end{bmatrix} + R_y\left({}^{I}\theta_G\right) R_z\left({}^{I}\psi_G\right) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi}_G \end{bmatrix} + R_y\left({}^{I}\theta_G\right) \begin{bmatrix} 0 \\ \dot{\theta}_G \\ 0 \end{bmatrix} $$

$$ {}^{G}\omega_{GI} = {}^{G}_{B}R \begin{bmatrix} p_B \\ q_B \\ r_B \end{bmatrix} + {}^{G}_{B}R \begin{bmatrix} 0 \\ 0 \\ \dot{\psi}_G \end{bmatrix} + R_y\left({}^{I}\theta_G\right) \begin{bmatrix} 0 \\ \dot{\theta}_G \\ 0 \end{bmatrix} \tag{11} $$

$$ {}^{I}\omega_{GI} = {}^{I}_{G}R\, {}^{G}\omega_{GI} = {}^{I}_{G}R \begin{bmatrix} p_G \\ q_G \\ r_G \end{bmatrix} \tag{12} $$

### c. *Gimbal Frame Angular Velocities with Respect to Inertia Frame*

If the camera frame is perfectly aligned with the gimbal platform frame, then there will be no rotation between the two frames. Therefore the angular rates between the two frames will be the same. The relationships are expressed below:

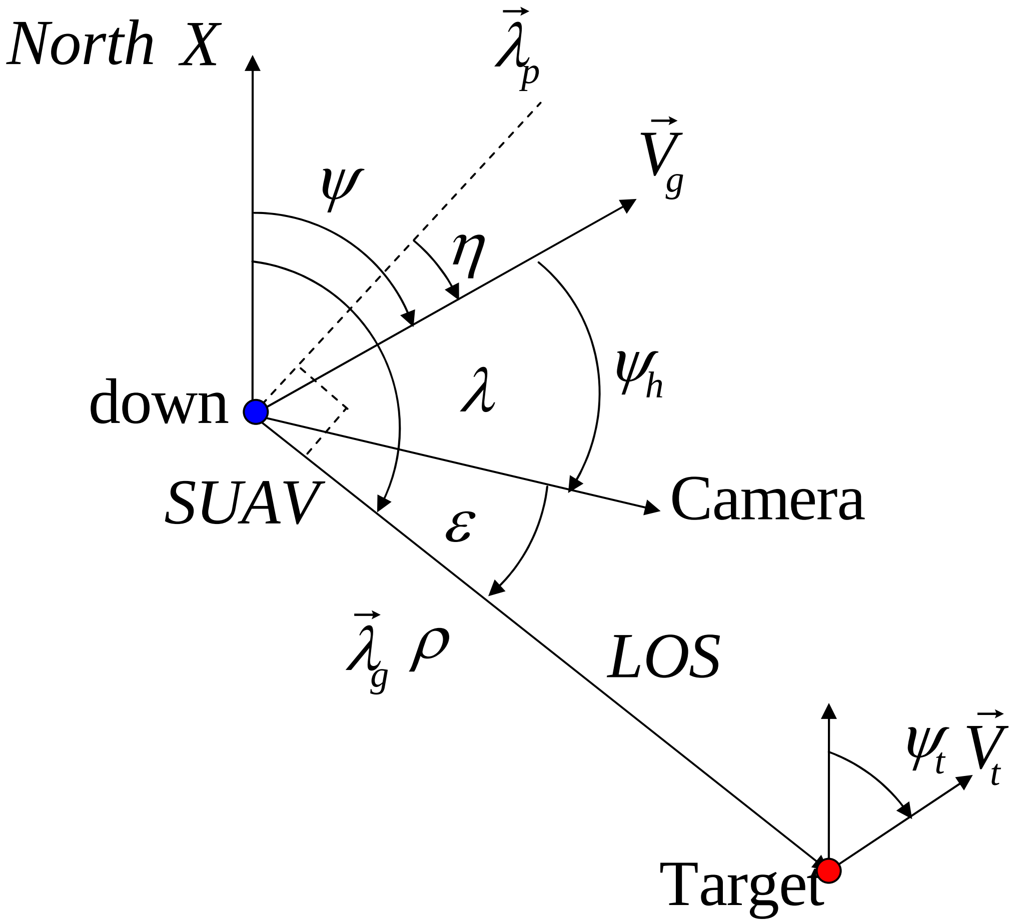
$$ {}^{C}\omega_{CI} = {}^{C}_{G}R\, {}^{G}\omega_{GI} \qquad ; {}^{C}_{G}R = I $$

$$ {}^{C}\omega_{CI} = {}^{G}\omega_{GI} = \begin{bmatrix} p_G \\ q_G \\ r_G \end{bmatrix} \tag{13} $$

$$ {}^{I}\omega_{CI} = {}^{I}\omega_{GI} \tag{14} $$

## D. KINEMATICS EQUATIONS OF THE SUAV-TARGET RELATIVE MOTION

A 2-D simplified representation of the kinematics relationships between the SUAV and target motion is required as a prerequisite to the development of the guidance control law as shown in Figure 6. It is assumed that an autopilot is capable of maintaining level flight of the UAV and the onboard 2-axis gimbal is inertially stabilized. This justifies the formulation of the target tracking and UAV control tasks in a horizontal, 2-D plane. The following figure depicts the kinematics relationships between the angles used to characterize relative motion of the SUAV-Target system [1], [5].




| | | | |
|---|---|---|---|
| $\eta$ | = navigation error | $\psi$ | =UAV heading in inertia frame |
| $\varepsilon$ | =camera LOS Pan error | $\psi_t$ | =Target heading in inertia frame |
| $\lambda$ | =LOS angle in inertia frame | $\psi_h$ | = Camera angle in UAV body frame |
| $\vec{\lambda}_g$ | = LOS vector | $\vec{V}_g$ | = SUAV ground speed in inertia frame |
| $\vec{\lambda}_p$ | = Normal to LOS vector | $\vec{V}_t$ | = Target ground speed in inertia frame |

Figure 6. Kinematics of the SUAV - Target Motion

The following set of basic kinematical relations is derived directly from the kinematics of Figure 6. It is first observed that:

$$\eta = \psi - (\lambda - \frac{\pi}{2}) \qquad (15)$$

Next, projecting the UAV and target speed vectors onto the LOS results in the time derivative of the horizontal range to the target

$$\dot{\rho} = -V_g \sin\eta + V_t \sin(\psi_t - (\psi - \eta)).$$

Similarly, projecting the same vectors onto the line orthogonal to the LOS produces the rotation speed of the LOS

$$\dot{\lambda} = \frac{V_g \cos\eta}{\rho} - \frac{V_t \cos(\psi_t - (\psi - \eta))}{\rho} \qquad (16)$$

Finally, an expression for the tracking error $\varepsilon$ is given by

$$\varepsilon = \lambda - \psi - \psi_h \qquad (17)$$

Substituting Equation (17) into time derivatives of Equations (15) and (16) produces the following set of equations describing the kinematics of the tracking problem:

$$\begin{aligned}
\dot{\eta} &= -\frac{V_g \cos\eta - V_t \cos(\psi_t - (\psi - \eta))}{\rho} + \dot{\psi} \\
\dot{\varepsilon} &= \frac{V_g \cos\eta - V_t \cos(\psi_t - (\psi - \eta))}{\rho} - \dot{\psi} - \dot{\psi}_h \\
\dot{\rho} &= -V_g \sin\eta + V_t \sin(\psi_t - (\psi - \eta))
\end{aligned} \qquad (18)$$

The navigation angle error $\eta$ is the angle subscribed between the SUAV's ground velocity vector and the perpendicular to LOS vector. The navigation angle error will approach zero when the SUAV has established a circular orbit about a stationary target at the desired range.

The camera LOS angle error $\varepsilon$ is the angle between the camera LOS vector (camera optical middle line) and the line connecting the camera and the

target. The objective of the gimbal control law is to drive this angle to zero by using a feedback of the image-based tracking software.

## E. CONTROL LAW DESIGN

The control objective is to drive $\eta$ and $\varepsilon$ to zero using the SUAV turn rate $\dot{\psi}$ and gimbal pan rate $\dot{\psi}_h$ commands as control inputs. To achieve this, the control laws for implementation on the SUAV autopilot controller and gimbal platform controller are as shown:

$$\begin{aligned} \dot{\psi} &= \frac{V_g}{\rho_d}\cos\eta - K_1\eta \\ \dot{\psi}_h &= K_1\eta + K_2\varepsilon \end{aligned} \qquad (19)$$

| | | | |
|---|---|---|---|
| $\eta$ | = Navigation angle error | $\dot{\psi}$ | = UAV turn rate in inertia frame |
| $\varepsilon$ | = Camera LOS angle error | $\dot{\psi}_h$ | = Camera turn rate in UAV body frame |
| $V_g$ | = SUAV ground speed | $\rho_d$ | = Desired range |
| $K_1, K_2$ | = Gain constants | | |

The chosen form of control law in Equation (19) allows for dynamic adjustment of the required turning bias of the SUAV, $\dot{\psi}$ in accordance to the magnitude of the ground velocity vector of the SUAV to converge to the desired range $\rho_d$. For a stationary target, if the SUAV starts tracking a target at a distance smaller than commanded range, it will spiral outwards to the desired range; if the SUAV starts tracking a target at a distance larger than the commanded range, it will spiral inwards to the desired range. When the SUAV flight is established at its desired range in a circular orbit, the turn rate of the SUAV will approach the required turning “bias” that will keep the SUAV in the circular orbit about the target at the commanded range. The gimbal turn rate of the gimbaled camera will also approach zero when the SUAV has established the +/- 90$^o$ camera LOS angle in the circular orbit at the desired range.

The feedback system consisting of Equations (18) and (19) is given by System (20) as follows:

$$
\begin{aligned}
&\dot{\eta} && = -V_g \rho_e \cos\eta - k_1\eta + V_t \cos(\eta-\psi)\overline{\rho} \\
&\dot{\varepsilon} && = V_g \rho_e \cos\eta - k_2\varepsilon - V_t \cos(\eta-\psi)\overline{\rho} \\
&\dot{\rho}_e && = \overline{\rho}^2 V_g \sin\eta - V_t \sin(\eta-\psi)\overline{\rho}^2 \\
&where && \rho_e = \frac{1}{\rho} - \frac{1}{\rho_d}; \quad \dot{\rho}_e = -\frac{1}{\rho^2}\dot{\rho}; \quad \overline{\rho} = \rho_e + \frac{1}{\rho_d}
\end{aligned} \qquad (20)
$$

From System (20), it can be observed that by driving the navigation angle error, $\eta$ and camera LOS angle error, $\varepsilon$ to zero, the range error $\rho_e$ (range to target) which is not observable but controllable, is indirectly driven to zero for a stationary target.

## F. LYAPUNOV STABILITY ANALYSIS OF THE FEEDBACK SYSTEM

For stability analysis, it is convenient to rescale $\eta$ by introducing a new state variable $\hat{\eta} = \eta/\rho_d$ and substituting into Equation (20):

$$
\dot{x} := \begin{bmatrix} \dot{\hat{\eta}} \\ \dot{\rho}_e \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} -V_g \rho_e \cos\hat{\eta}\rho_d - k_1\hat{\eta}\rho_d + V_t \cos(\hat{\eta}\rho_d - \psi)\overline{\rho} \\ \overline{\rho}^2 V_g \sin\hat{\eta}\rho_d - V_t \cos(\hat{\eta}\rho_d - \psi)\overline{\rho}^2 \\ V_g \rho_e \cos\hat{\eta}\rho_d - k_2\varepsilon - V_t \cos(\hat{\eta}\rho_d - \psi)\overline{\rho} \end{bmatrix} \qquad (21)
$$

$$
where \quad x = \begin{bmatrix} \hat{\eta} & \rho_e & \varepsilon \end{bmatrix}^T
$$

Using eigenvalue analysis, the stability of the system in Equation (18) is addressed next. If we first assume that the target is stationary and therefore $V_t = 0$ then:

$$
\dot{x} := \begin{bmatrix} \dot{\hat{\eta}} \\ \dot{\rho}_e \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} -V_g \rho_e \cos\hat{\eta}\rho_d - k_1\hat{\eta}\rho_d \\ \overline{\rho}^2 V_g \sin\hat{\eta}\rho_d \\ V_g \rho_e \cos\hat{\eta}\rho_d - k_2\varepsilon \end{bmatrix}, \qquad (22)
$$

and the origin $x = (0\ 0\ 0)$ is clearly the equilibrium of Equation (19). Linearization of Equation (22) around the origin yields an LTI system:

$$\dot{\xi} = \begin{bmatrix} -k_1 & -V_g / \rho_d & 0 \\ V_g / \rho_d & 0 & 0 \\ 0 & V_g & -k_2 \end{bmatrix} \xi \tag{23}$$

We may also assume also that the SUAV velocity $V_g$ is constant and $V_g \in \left[V_{g_{\min}}, V_{g_{\max}}\right], V_{g_{\max}} \geq V_{g_{\min}} > 0$. Then the eigenvalues of the state matrix in Equation (23) have negative real parts for any $k_1 > 0,\ k_2 > 0$. Therefore, the nonlinear system in Equation (22) is locally asymptotically stable for any $k_1 > 0,\ k_2 > 0$.

On the other hand, if target is moving and $V_t \neq 0$, the equilibrium of Equation (21) is at the relative heading $\psi = \frac{\pi}{2}$. This in fact corresponds to the circular motion of the SUAV around the target. In this case linearization of Equation (21) around the equilibrium results in an LTI system

$$\dot{\xi}_1 = \begin{bmatrix} -k_1 + V_t / \rho_d & -V_g / \rho_d & 0 \\ \left(V_g - V_t\right) / \rho_d & 0 & 0 \\ -V_t & V_g & -k_2 \end{bmatrix} \xi_1 \tag{24}$$

The target velocity $V_t$ is assumed constant and $V_t \in \left[V_{t_{\min}}, V_{t_{\max}}\right], V_{t_{\max}} \geq V_{t_{\min}} > 0$. The eigenvalues of the LTI system in Equation (21) will have negative real parts if $V_g > V_t$, $k_1 > V_t / \rho_d$ and $k_2 > 0$.

The plots of the state trajectories of the system in Equation (20) in response to a number of initial conditions are included as follows:

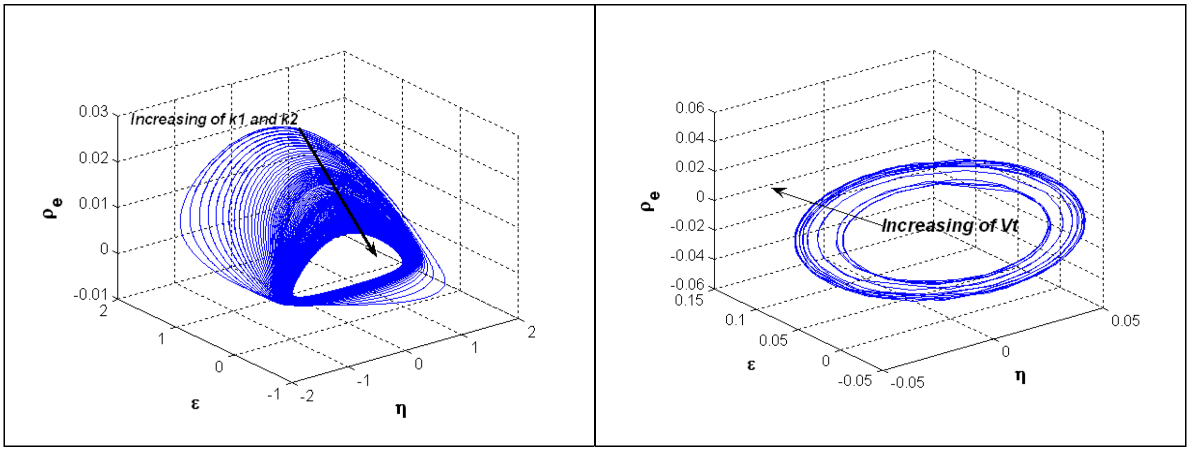



Figure 7. Steady State Trajectories

In Figure 7, the plot on the left illustrates the impact of increasing the gains $k_1$, $k_2$.for a fixed $V_t$ on the trajectories of the feedback system in Equation (20). Conversely, the plot on the right demonstrates the influence of increasing $V_t$ for fixed $k_1$ and $k_2$. The figures show that the trajectories of feedback system in Equation (20) converge to a ball, the size of which is proportional to $V_t$ and inversely proportional to $k_1$ and $k_2$ [1].

# III. IMPLEMENTATION OF CONTROL LAW

## A. CONTROL SYSTEM ARCHITECTURE

The control law that will be implemented is shown in Equation (25) and the control system architecture that implements the control laws is presented in Figure 8.

$$\begin{aligned} \dot{\psi} &= \frac{V_g}{\rho_d}\cos\eta - K_1\eta \\ \dot{\psi}_h &= K_1\eta + K_2\varepsilon \end{aligned} \qquad (25)$$

Onboard cameras provide real-time imagery to the image tracking software and the software computes the tracking error $\varepsilon$, while the onboard GPS and inertial systems provide solution for the navigation error $\eta$. In turn, $\dot{\psi}$ and $\dot{\psi}_h$ are computed and these yaw rates drive the autopilot and the gimbal respectively.

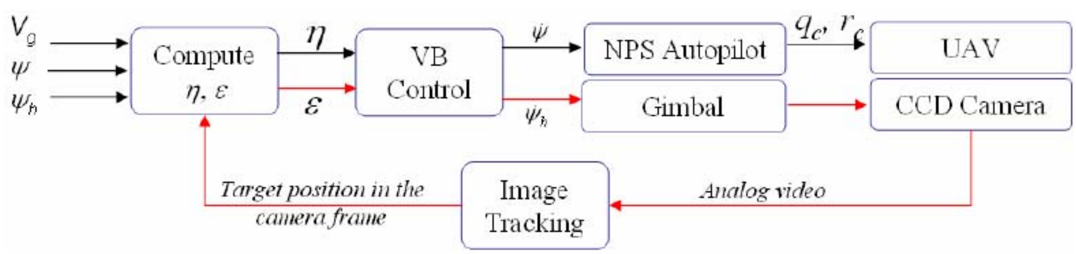



Figure 8. Control System Architecture

The control law for the gimbal camera yaw command $\dot{\psi}_h$ in Equation (25) is implemented in the SIMULINK between the “Gimbaled Camera model” and “CurGuid Controller” block [5] as shown in Figure 9.

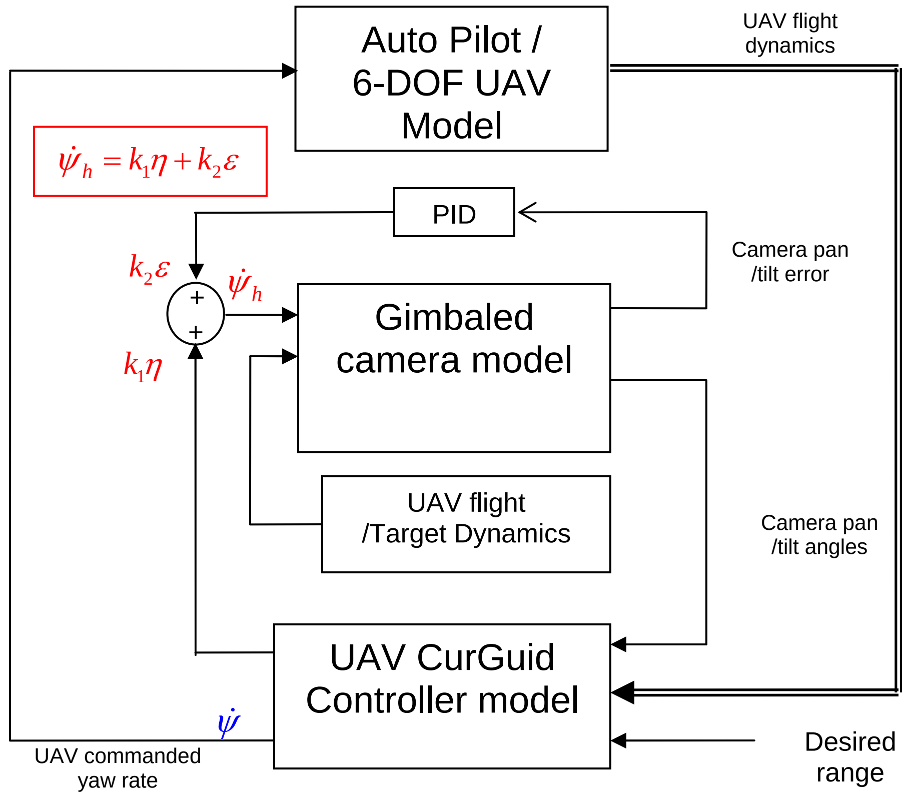



Figure 9. VBTT SIMULINK Model Schematics

The control law for the SUAV yaw rate command $\dot{\psi}$ in Equation (25) is implemented in the "CurGuid Controller" block, as shown in Figure 10. in the VBTT SIMULINK model schematics [5].

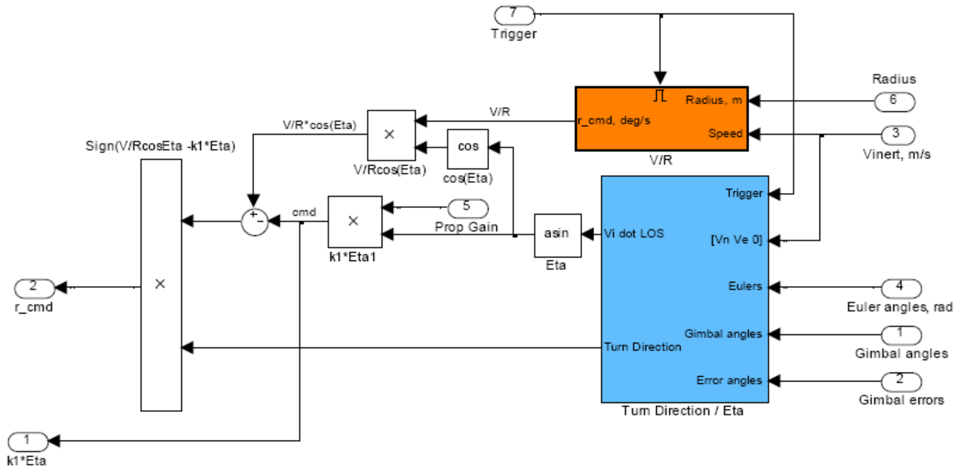



Figure 10. SIMULINK CurGuid Controller Block

## B. SIMULATION RESULTS

The performance of the existing control law was previously tested against a stationary target and a moving target under different target motion scenarios [5]. From these experiments, the sensitivity results of the new control law are derived for various parameters of feedback control gain $K_1$, and the relative velocity ratio of the target and SUAV velocities, $\frac{V_t}{V_g}$.

The analysis presented herein generalizes the previous results and further investigate the robustness of the control law to the variation of control gain $K_1$ with different target maneuver scenarios. As the parameter $K_2$ only partially affects the gimbal control law equation ($\dot{\psi}_h = k_1\eta + k_2\varepsilon$), $K_2$ is not examined for the cross coupled sensitivity between the SUAV turn rate command and the camera LOS turn rate command. The scenarios used for analysis are as follows:

- **Scenario A:** Stationary target.
- **Scenario B:** Target is moving with a nominal speed (8m/s) with a constant heading.
- **Scenario C:** Target is moving with a nominal speed (8m/s)[2] with variable heading.

In addition, the three major error components of a SUAV tracking performance namely: navigation error, $\eta$, camera LOS yaw error, $\varepsilon$ and relative range holding error, $\rho^*$ are numerically calculated and analyzed. Relative range holding error, $\rho^*$ is a measure of the SUAV range holding capability and is defined by the ratio of the range error, $\rho_{err}$ to the range desired, $\rho_d$. These three error components are assumed independent of each other. Their steady state time averaged values are measured for different values of feedback gain, $K_1$. It is therefore desired to obtain a value of $K_1$ such that it minimizes all the error components simultaneously. For the purpose of multi-criteria optimization, a new measure of performance (MOP), $M_1$ is introduced as a square root of the sum squares of the three independent error components. This choice of the MOP is motivated by the fact that each $K_1$ can be represented by a point in a 3-dimensional (3-D) space of errors $\{\eta, \rho^*, \varepsilon\}$ with a control objective to minimize all of them simultaneously. This corresponds to the search for a $K_1$ such that the distance from the origin of the 3-D errors space is minimized as graphically shown in Figure 11. This can be represented for a hypothetical distribution of $\{\eta, \rho^*, \varepsilon\}$ versus $K_1$ as shown in Equation (26):

[2] $V_t$ = 8m/s is chosen because it exhibits sensitivity to the different target maneuver scenarios.

$$M_1 = \sqrt{\eta^2 + \varepsilon^2 + \rho^{*2}} \quad (26)$$

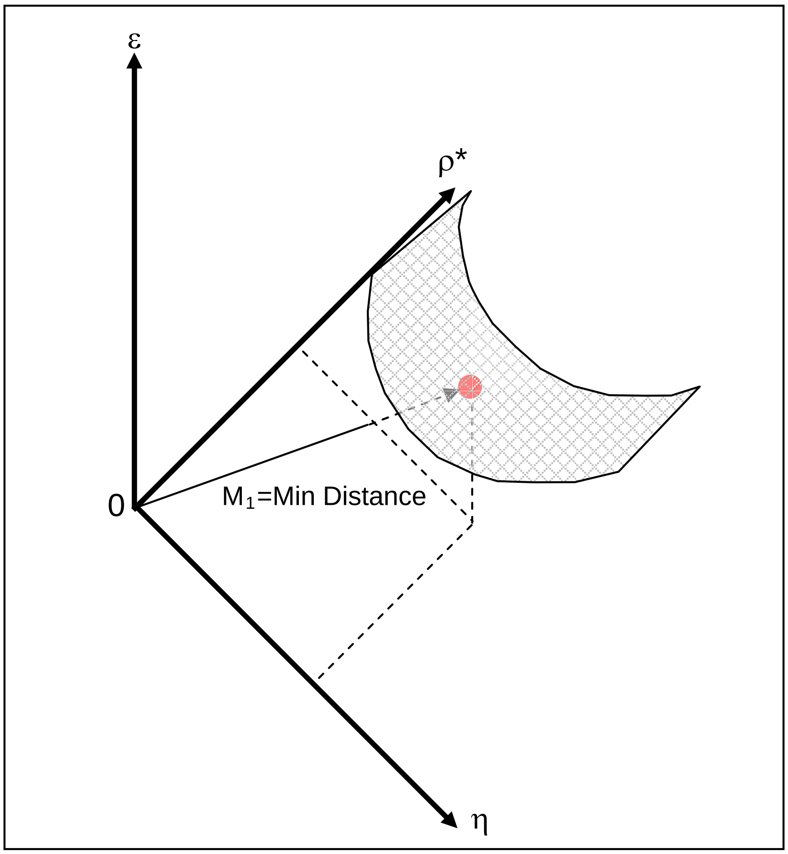



Figure 11. Intuitive Motivation of Multi Objective Optimization of Three Independent Criteria {η, ρ*,ε}

Finally, the robustness of the SUAV control law is also investigated by considering generalized time delay in the feedbacks due to for example data processing or communications lag. Therefore the target maneuver scenarios are extended as follows:

- o **Scenario A1:** Target is stationary with different values of time delay.
- o **Scenario C1:** Target is moving with a nominal speed (8m/s) with varying heading and different values of time delay.

### 1. Sensitivity Analysis of Measure of Performance to Variations of Gain $K_1$ for Scenario A: Stationary Target

The sensitivity analysis of the measure of performance, $M_1$ to variations of gain $K_1$ is examined in a scenario where the target is stationary and the commanded range is 300m. This scenario is chosen as a baseline model for further comparison with other target maneuver scenarios. The initial conditions are: (1) SUAV velocity = 28 m/s; (2) target velocity = 0m/s; (3) initial position of the SUAV is at [0, -1000, 300]; (4) initial position of the target is at [0, 0, 0] and (5) $K_2$ = 0.20. Figure 12 shows the variation of $\eta$, $\rho^*$, $\varepsilon$ and $M_1$ with variations of gain $K_1$ for Scenario A while Figure 13 shows the error dynamics for the optimal case of $K_1$ = 0.500.

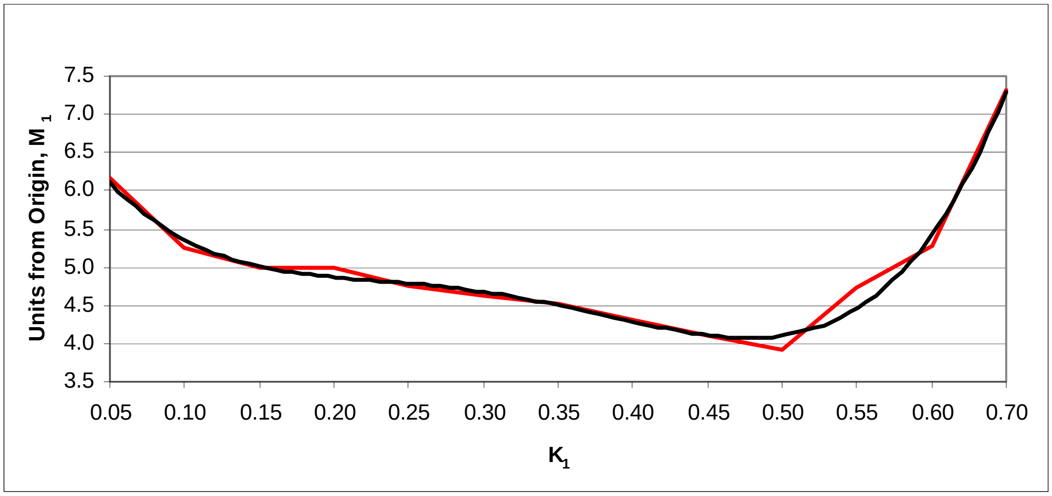
The following observations can be made from the plots:

(a) From Figure 12, it is observed that the relative range holding error, $\rho^*$ increases with increasing $K_1$. This is accompanied by a corresponding decrease in the $\eta$ and $\varepsilon$ errors, resulting in an overall decrease of $M_1$.

(b) From Figure 13, it follows that the increasing of $K_1$ extends the settling time.

Scenario A: Stationary Target

Measure of Performance, $M_1$ with Variations in $K_1$




Performance of $\eta$, $\rho^*$, $\varepsilon$ and $M_1$ with Variations in $K_1$

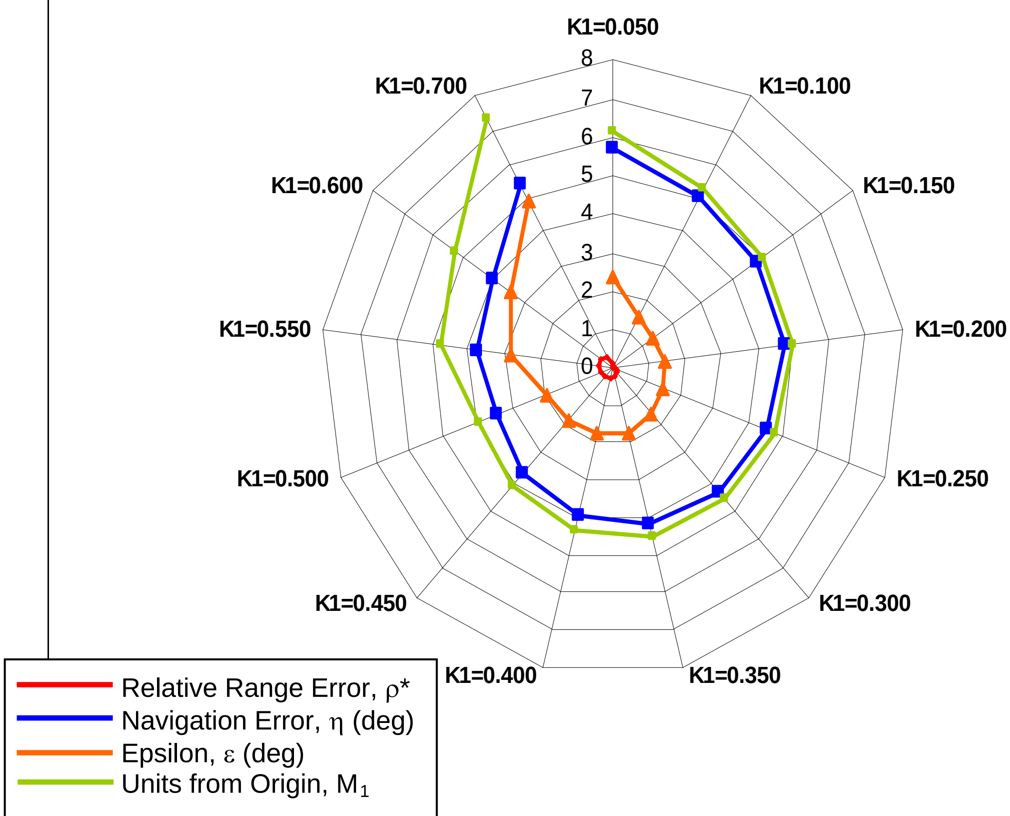
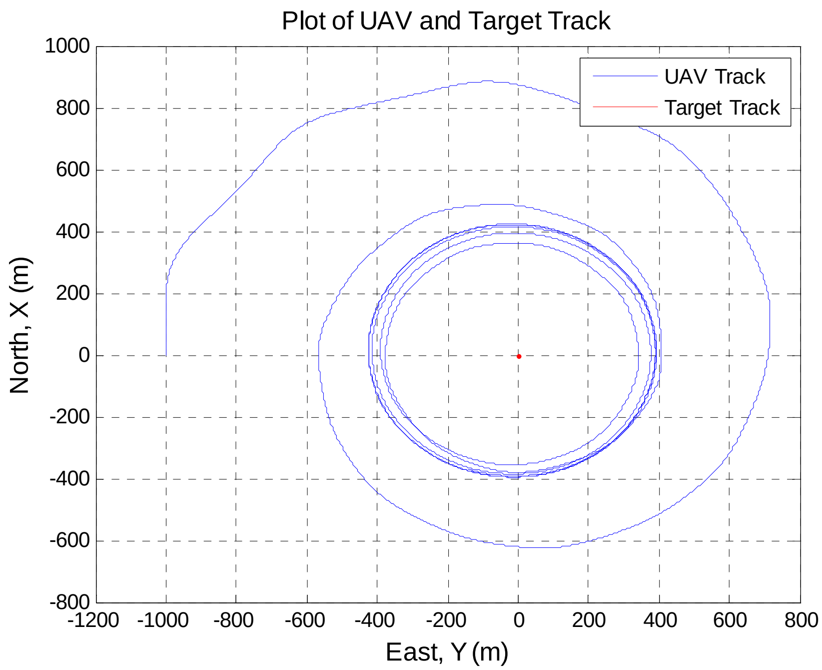


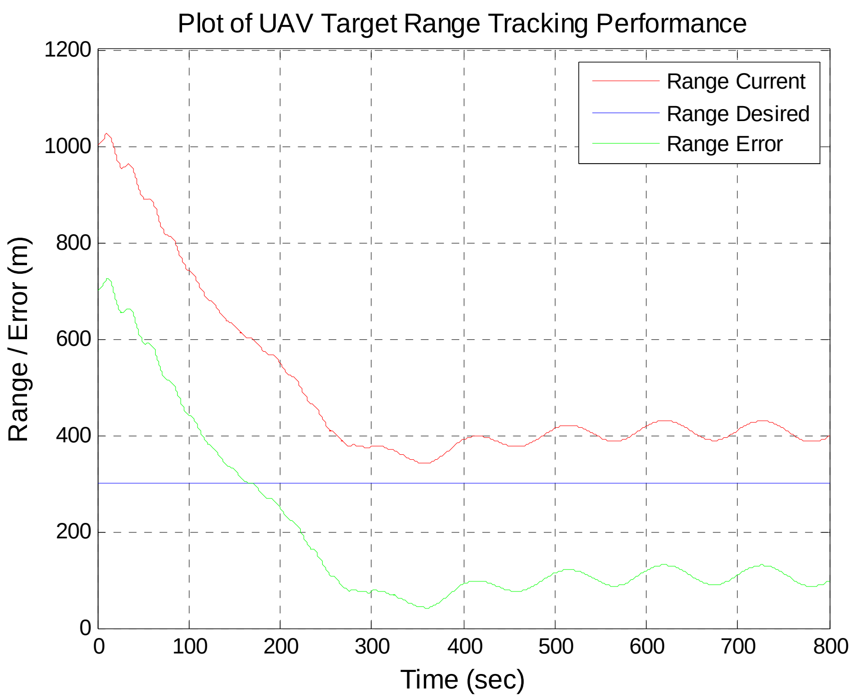

Figure 12. Variation of $M_1$ vs $K_1$ for Scenario A

Scenario A: $K_1$ = 0.500 (Optimal Case)

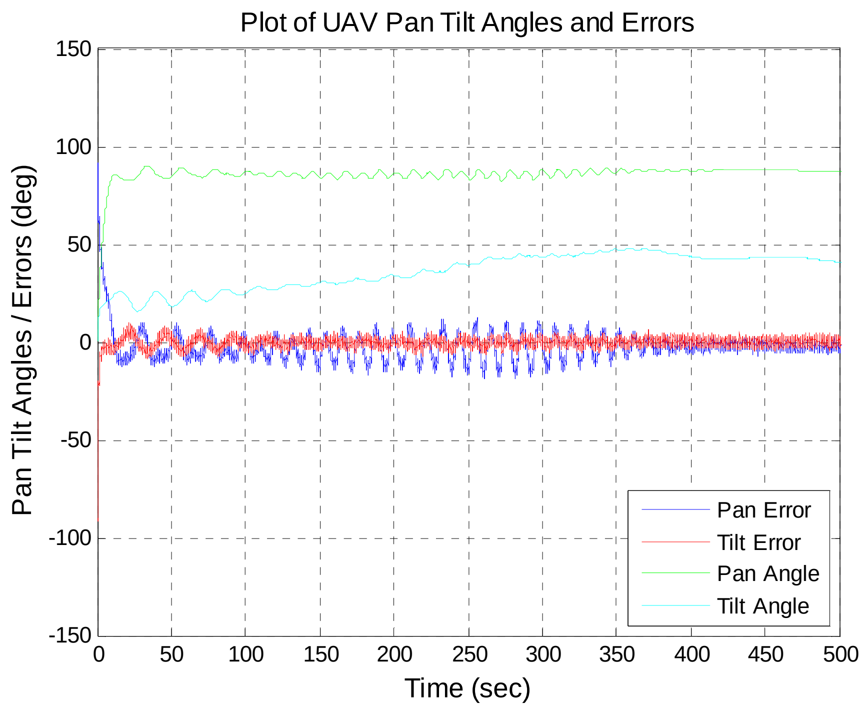
| SUAV target VBTT control law tracking history |
|---|




| Range convergence history of the SUAV to the desired range of 300m |
|---|



Pan / Tilt angles convergence history and gimbal camera angle errors




Navigation angle error convergence history

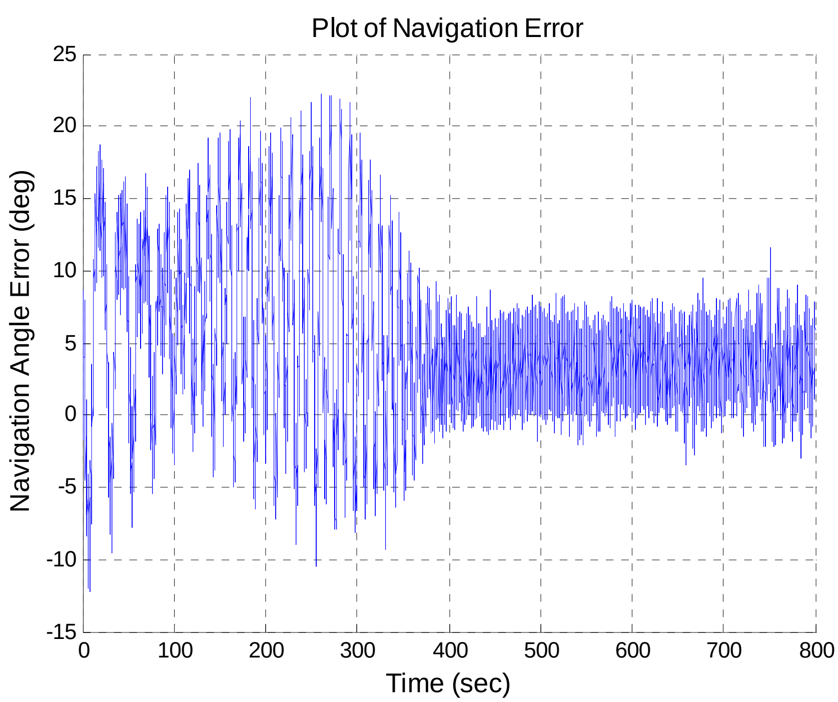



Figure 13. Sensitivity Analysis for $K_1$ = 0.500 for Scenario A

### 2. Sensitivity Analysis of Measure of Performance to Variations of $K_1$ for Scenario B: Target is Moving with a Nominal Speed (8m/s) with a Constant Heading

The sensitivity analysis of the measure of performance, $M_1$ to variations of parameter $K_1$ is examined in a scenario where the target is moving with constant heading and the commanded range is 300m. This scenario investigates the effects of constant target heading on the errors $\eta$, $\rho$*, $\varepsilon$ and $M_1$. The initial conditions are: (1) SUAV velocity = 28 m/s; (2) target velocity = 8m/s; (3) initial position of the SUAV is at [0, -1000, 300]; (4) initial position of the target is at [0, 0, 0] and (5) $K_2$ = 0.20.

The same nominal value of 8m/s is chosen to investigate the effect of constant target heading. Figure 14 shows the variation of $\eta$, $\rho$*, $\varepsilon$ and $M_1$ with variations in $K_1$ for Scenario B while Figure 15 shows the error dynamics for the optimal case of $K_1$ = 0.550 for Scenario B.

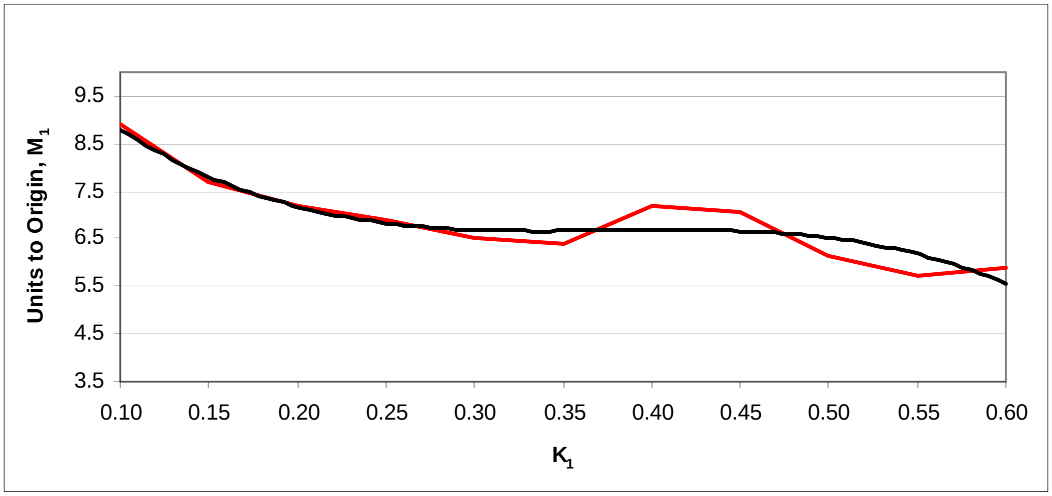
The following observations can be made from the plots:

(a) From Figure 14, It is observed that the relative range holding error, $\rho$* increases with increasing $K_1$. This is accompanied by a corresponding decrease in the $\eta$ and $\varepsilon$ errors, resulting in an overall decrease of $M_1$.

(b) From Figure 15, it follows that the increasing of $K_1$ extends the settling time.

Scenario B: Moving Target with Constant Target Heading

| Measure of Performance, $M_1$ with Variations in $K_1$ |
|---|




| Performance of η, ρ*, ε and $M_1$ with Variations in $K_1$ |
|---|

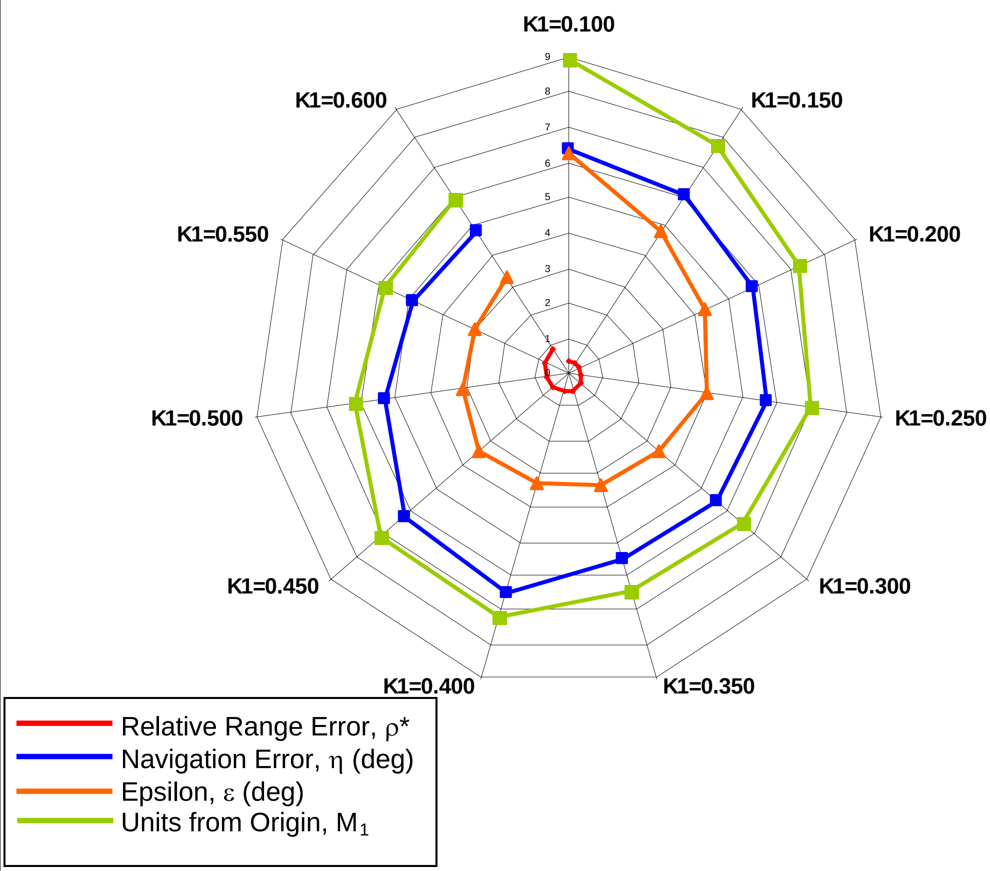
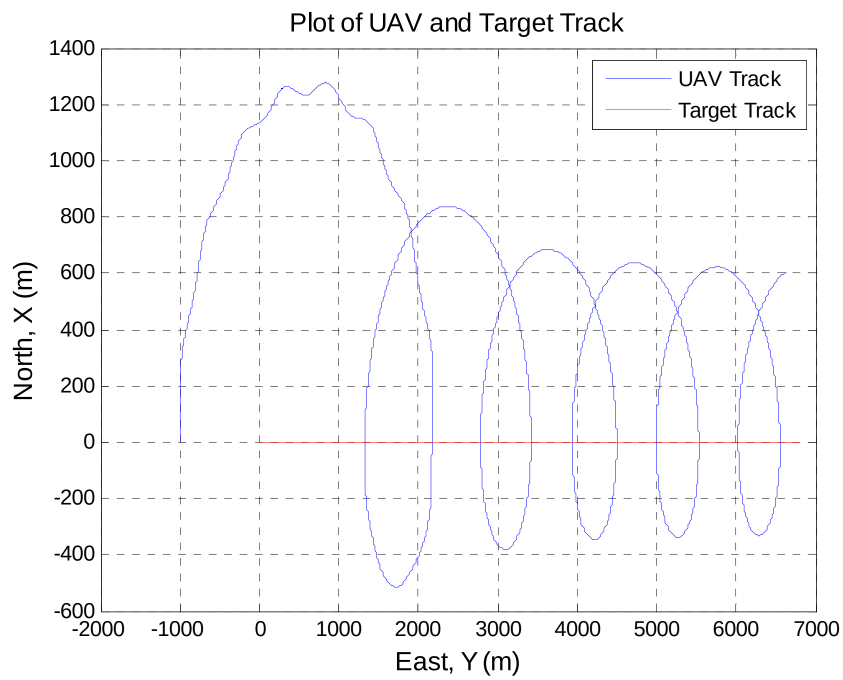
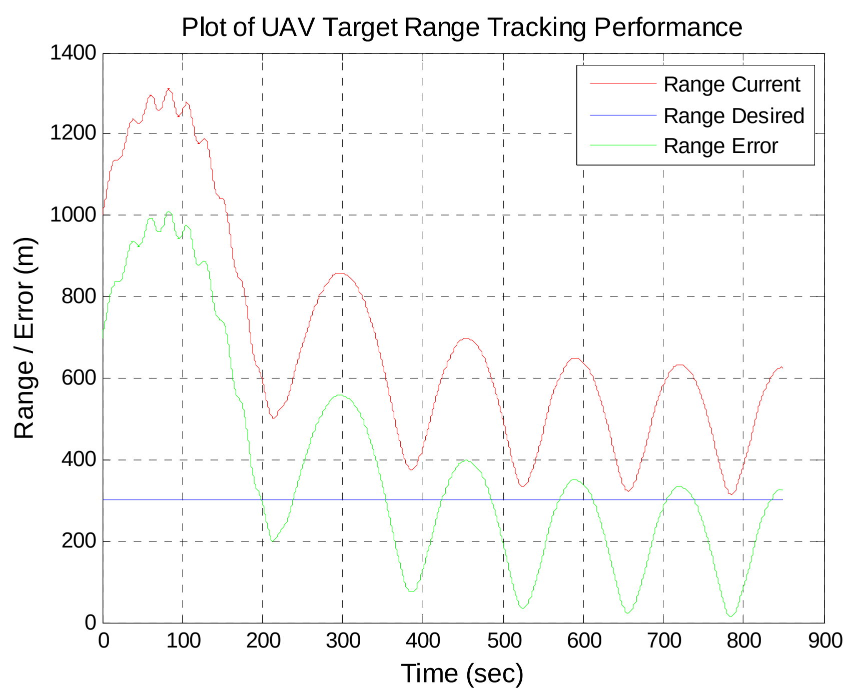



Figure 14. Variation of $M_1$ vs $K_1$ for Scenario B

Scenario B: $K_1$ = 0.550 (Optimal Case)

| SUAV target VBTT control law tracking history |
|---|
|  |
| Range convergence history of the SUAV to the desired range of 300m |
|  |

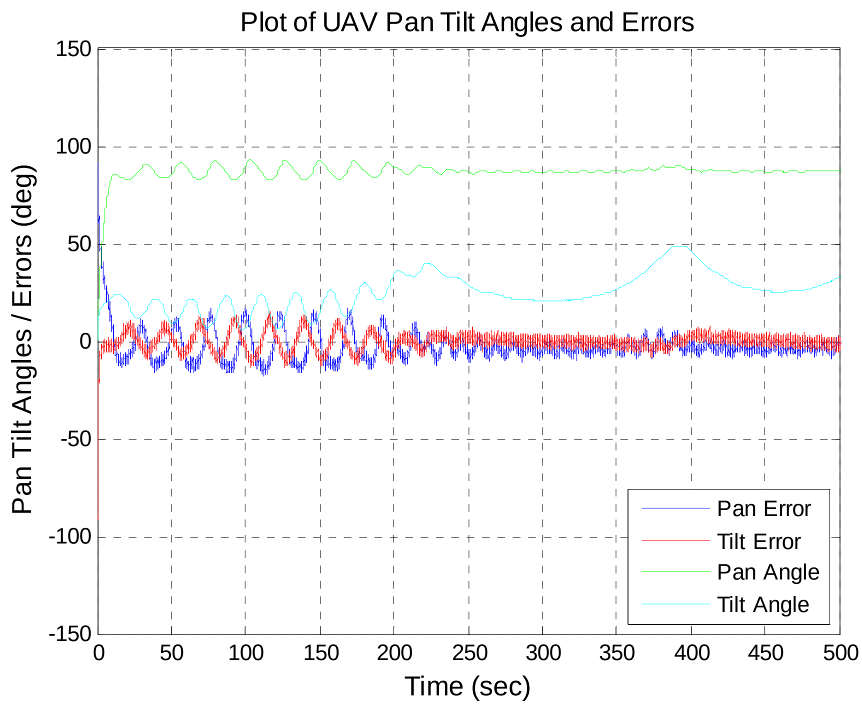
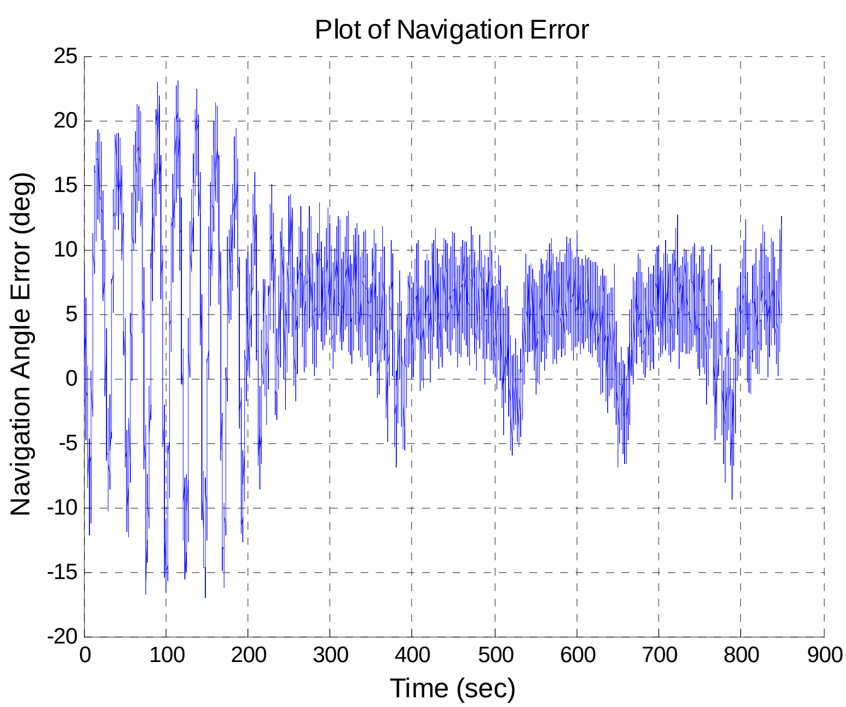
| Pan / Tilt angles convergence history and gimbal camera angle errors |
| --- |
|  |
| Navigation angle error convergence history |
|  |





Figure 15. Sensitivity Analysis for $K_1$ = 0.550 for Scenario B

### 3. Sensitivity Analysis of Measure of Performance to Variations of $K_1$ for Scenario C: Target is Moving with a Nominal Speed (8m/s) with Variable Heading

The sensitivity analysis of the measure of performance, $M_1$ to variations of parameter $K_1$ is examined in a scenario where the target is moving with varying target heading and the commanded range is 300m. This scenario investigates the effect of variable target heading on the errors $\eta$, $\rho^*$, $\varepsilon$ and $M_1$. The initial conditions are: (1) SUAV velocity = 28 m/s; (2) target velocity = 8m/s; (3) initial position of the SUAV is at [0, -1000, 300]; (4) initial position of the target is at [0, 0, 0] and (5) $K_2 = 0.20$.

Previous results obtained [5] indicated that the SUAV's range holding capability decreases as the target speed increases from 5m/s to 15m/s. Therefore, a nominal value of 8m/s is chosen to investigate the effect of varying target heading. Figure 16 shows the variation of $\eta$, $\rho^*$, $\varepsilon$ and $M_1$ with variations in $K_1$ for Scenario C while Figure 17 shows the error dynamics for the optimal case of $K_1 = 0.550$.

The following observations can be made from the plots:

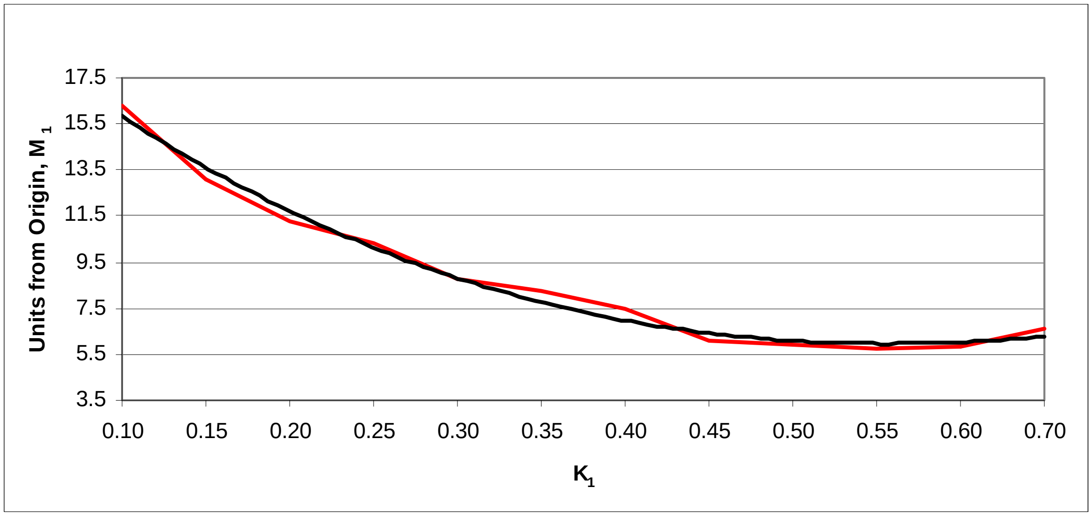
(a) From Figure 16, it is observed that the relative range holding error, $\rho^*$ increases with increasing $K_1$. This is accompanied by a corresponding decrease in the $\eta$ and $\varepsilon$ errors, resulting in an overall decrease of $M_1$.

(b) From Figure 17, it follows that the increasing of $K_1$ extends the settling time.

(c) Compared to the stationary target scenario, the value of optimal $K_1$ increases when there is a disturbance to the target motion. When the feedback gain K increases, the control effort required to compensate for errors usually increases correspondingly.

Scenario C: Moving Target with Variable Target Heading

Measure of Performance, $M_1$ with Variations in $K_1$




Performance of $\eta$, $\rho^*$, $\varepsilon$ and $M_1$ with Variations in $K_1$

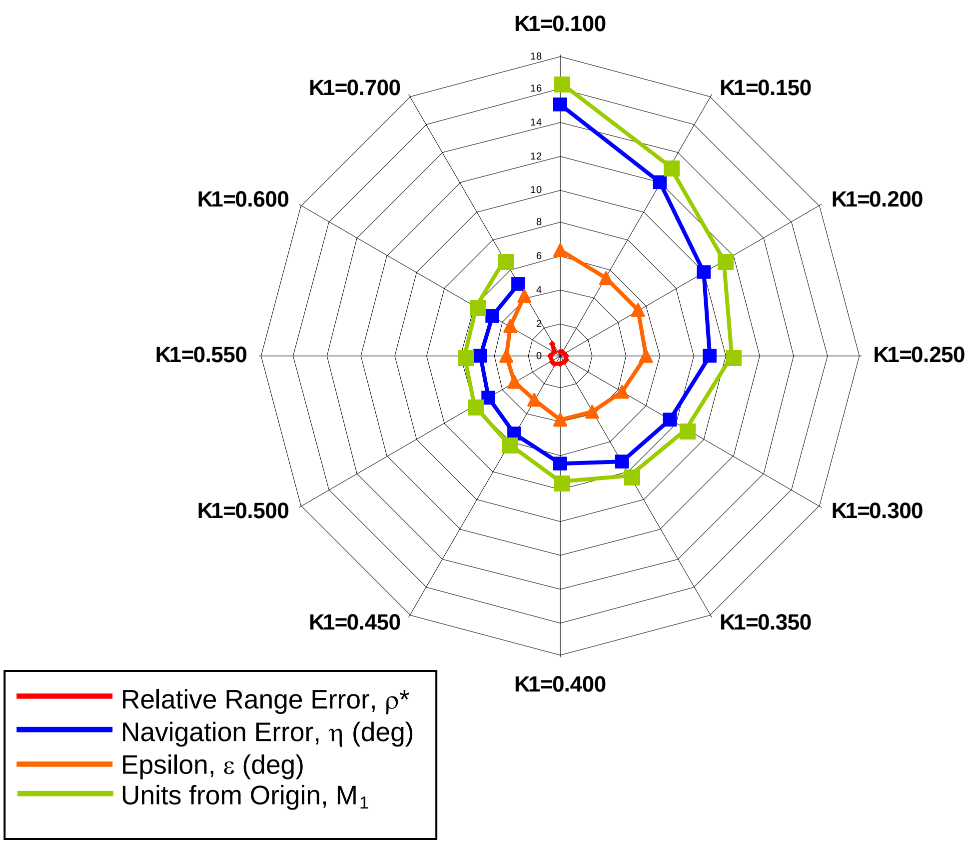
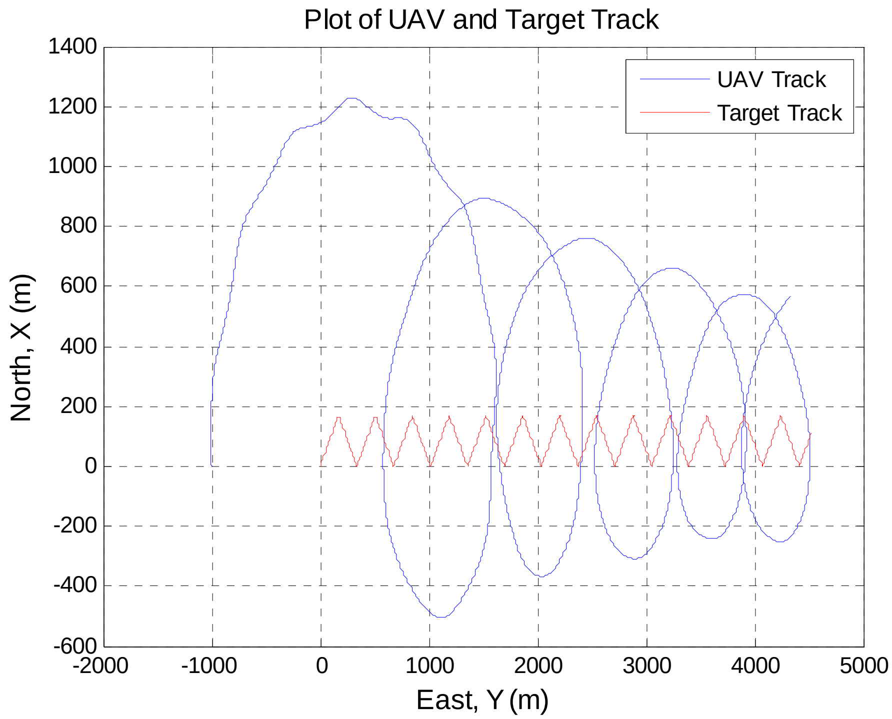


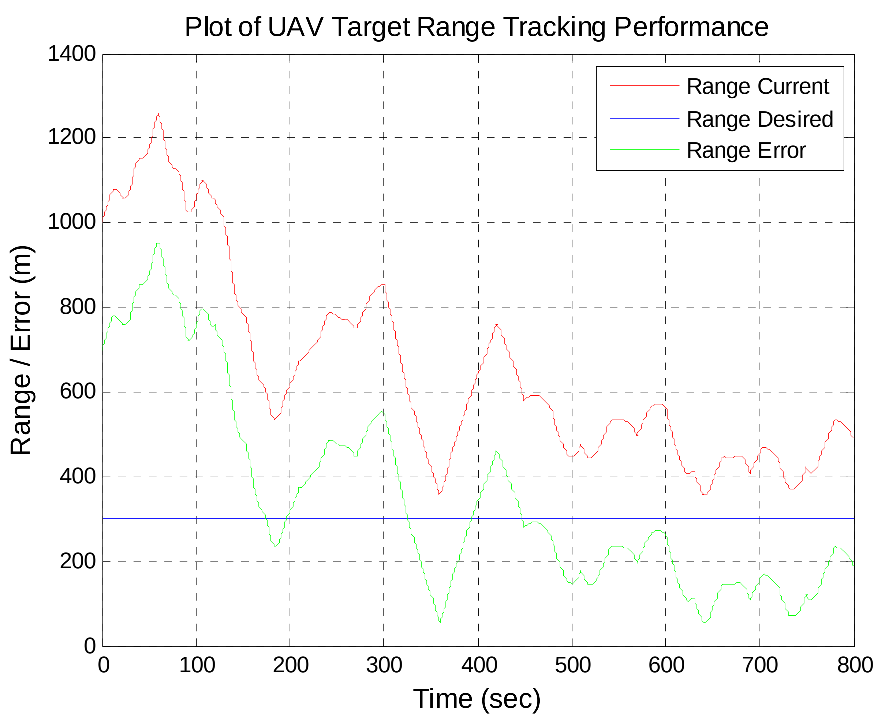

Figure 16. Variation of $M_1$ vs $K_1$ for Scenario C

Scenario C: $K_1$ = 0.550 (Optimal Case)

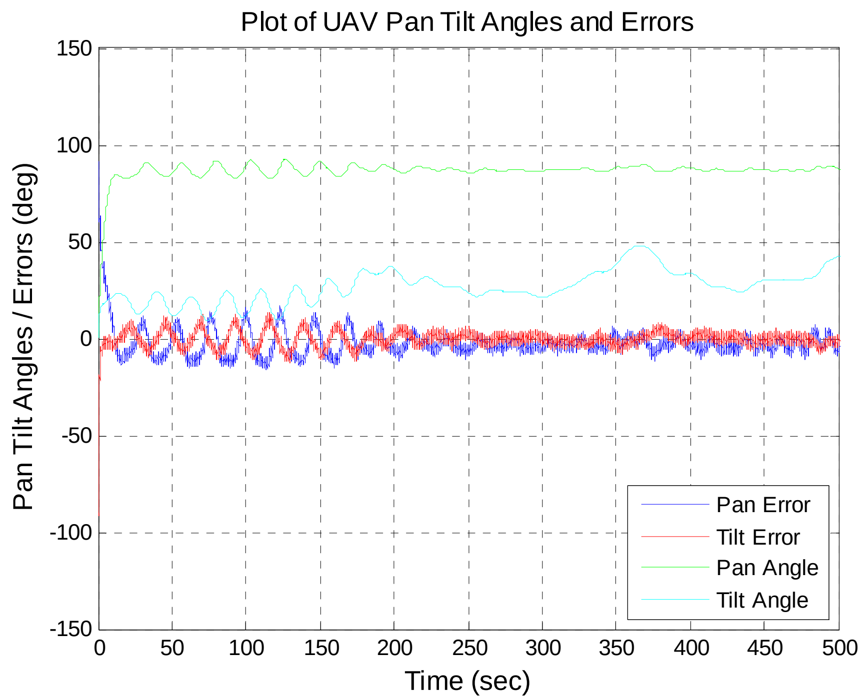
SUAV target VBTT control law tracking history




Range convergence history of the SUAV to the desired range of 300m



Pan / Tilt angles convergence history and gimbal camera angle errors




Navigation angle error convergence history

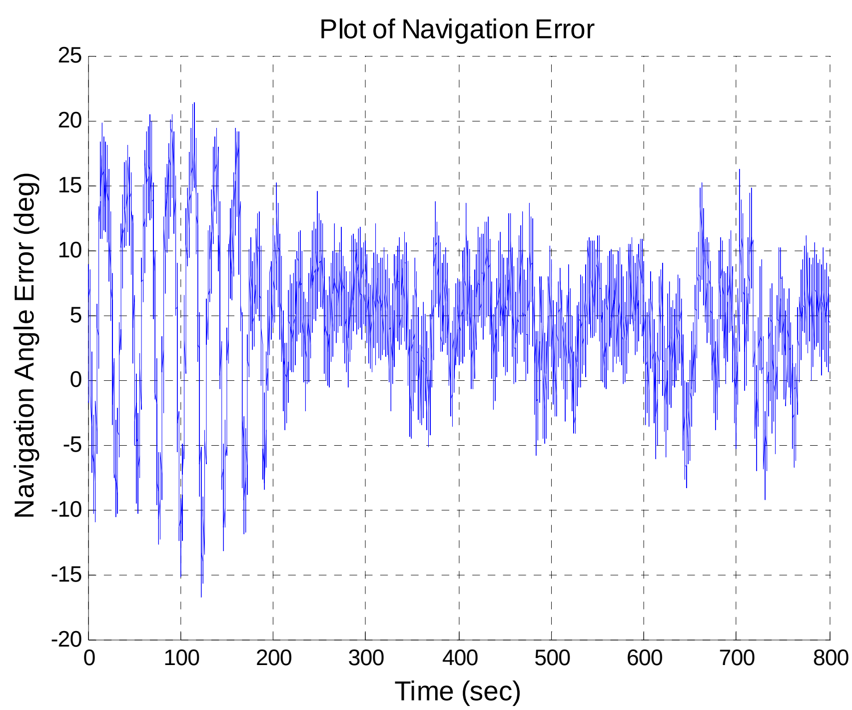



Figure 17. Sensitivity Analysis for $K_1$ = 0.550 for Scenario B

### 4. Summary of Simulation Results

#### *a. Variations of $M_1$ for Scenarios A, B, C*

Figure 18 summarizes the trend of $M_1$ with variations in $K_1$ for the three target maneuver scenarios. It is observed that for all values of $K_1$, the range of values for $M_1$ is minimal in a stationary target scenario while the $M_1$ values in a variable target heading scenario (Scenario C) are the largest. Moreover, the variation of the target heading has a more adverse impact on the $M_1$ values compared to the target speed. It is also observed that $M_1$ is minimal at a value of K1 = 0.50 for Scenario A while $M_1$ is minimal at a value of $K_1$ = 0.550 for Scenario B and C. Finally, it is noted that the region of optimal $K_1$ is bounded with small variations of feasible $K_1$. This allows the optimal value for all three scenarios to be quickly found by further analysis within the bounded feasible region of $K_1$.

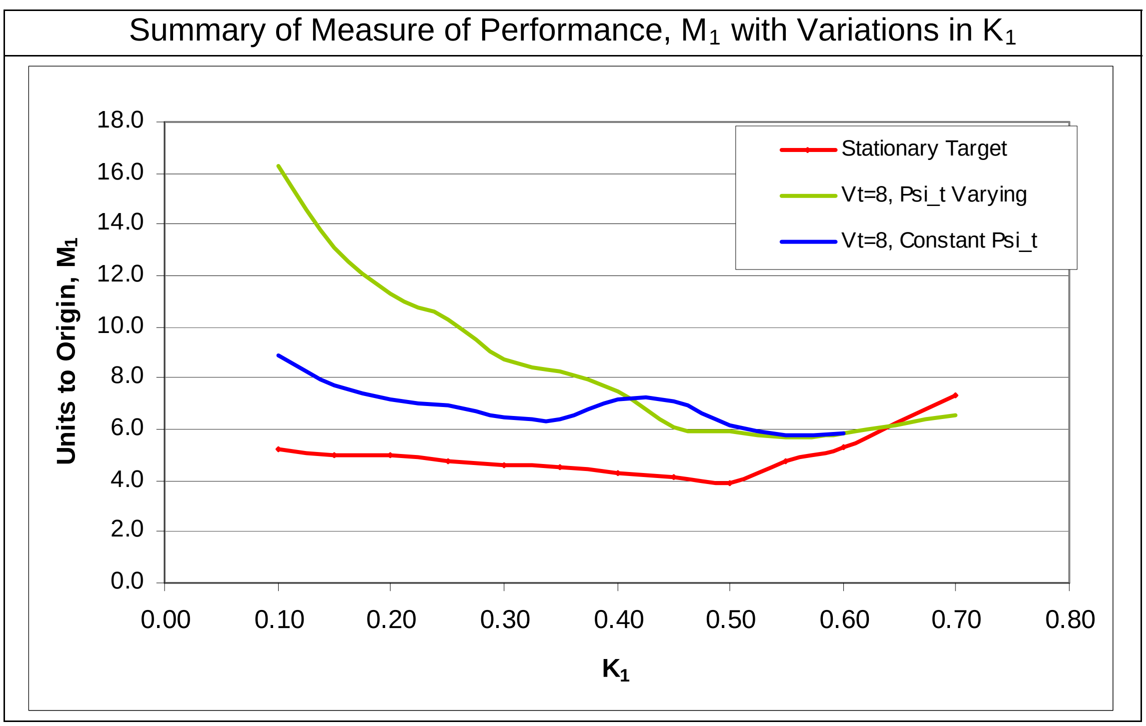



Figure 18. Plot of $M_1$ vs $K_1$ for Scenarios A, B, C

### b. *Variations of $M_1$ for Scenario A1: Target is Stationary with Different Values of Time Delay*

Figure 19 illustrates the impact on $M_1$ with the introduction of a time delay to the system. It is observed that as the value of the time delay increases, it significantly limits the feasible values of $K_1$. As the value of time delay increases, $M_1$ increases exponentially (loss of optimality).

For each plot, the departure point where there is a significant change in the slope (denoted by a black circle) is analyzed. It can be observed that the black circles and hence the value of $M_1$ increases as the time value is increases. In conclusion, the introduction of time delay requires a significant decrease of $K_1$. This results in a significant loss of optimality, increasing the overall error.

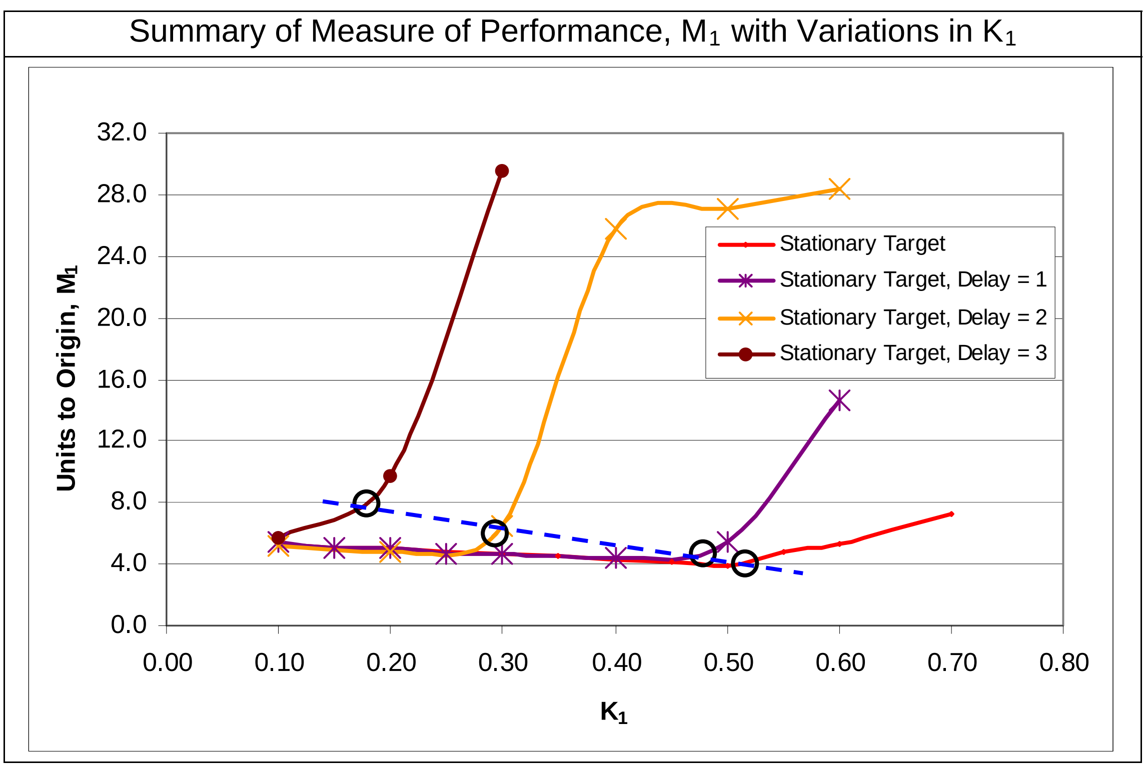



Figure 19. Plot of $M_1$ vs $K_1$ for Scenario A1.

### c. *Variations of $M_1$ for Scenario C1: Target is Moving with a Nominal Speed (8m/s) with Variable Heading with Different Values of Time Delay*

Figure 20 illustrates the impact on $M_1$ with the introduction of a time delay to the system. It is observed that as the value of the time delay increases, it limits the feasible values of $K_1$. As the value of time delay increases, $M_1$ increases exponentially (loss of optimality) with increasing values of $K_1$.

For each plot, the departure point where there is a significant change in the slope (denoted by a black circle) is analyzed. It can be observed that the black circles and hence the value of $M_1$ increases as the time delay is increases. In conclusion, the introduction of time delay requires a significant decrease of $K_1$. This results in a significant increase in the overall error.

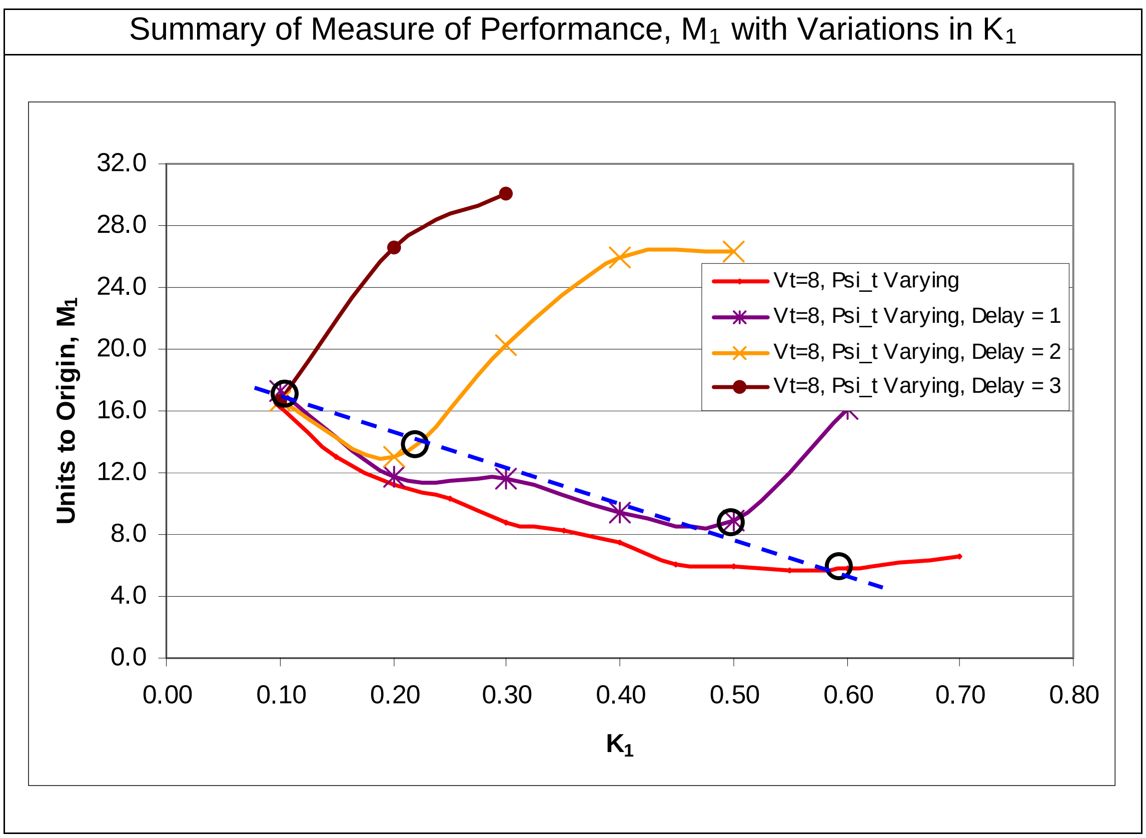


Figure 20. Plot of $M_1$ vs $K_1$ for Scenario C1

## C. CONCLUSIONS ON SIMULATION RESULTS

The SIMULINK based results obtained in the preceding section investigate the various target maneuver scenarios and the effect of the measure of performance $M_1$, with variation in the feedback gain, $K_1$. These results showed that there is no single value of $K_1$ that is able to satisfy all scenarios. This implies that the existing control law is not robust enough to meet different operational scenarios and therefore it is necessary to develop an improved control law. A new adaptive vision based guidance law is recommended for this purpose.

## D. DISCUSSION OF NEW ADAPTIVE VISION BASED GUIDANCE LAW

The new adaptive vision based guidance law [6] derives the estimates of the target's velocity by using a fast estimation algorithm with analytically proven performance bounds. A review of the target motion estimation developments is presented below.

Let $p(t)=[p_x(t),p_y(t),p_z(t)]^T$ be the position of the target with respect to the UAV in the inertial frame and let h(t) denote the relative altitude of the UAV above the target. Let $V_{UAV}(t)$ be the UAV's speed and let $V_g(t)$ be the projection of $V_{UAV}(t)$ onto the horizontal plane. Denoting the UAV flight path angle by $\gamma(t)$, $V_g(t)=V_{UAV}(t)\cos\gamma(t)$. Let $V_t(t)$ and $\psi_t(t)$ be the amplitude and the orientation of the target's velocity in the horizontal plane and $V_h(t)$ be the rate of change of target elevation. Let $\eta(t)$ denote the angle between the UAV's velocity vector and the vector perpendicular to the light of sight (LOS). The kinematic equations for a UAV tracking a target are:

$$\dot{\rho}(t) = -V_g(t)\sin\eta(t) + V_t(t)\sin(\psi_t(t) - (\psi(t) - \eta(t))), \qquad \rho(0) = \rho_0,$$
$$= \beta_1(\omega(t))\sin(\eta(t) + \beta_1(\omega(t))),$$

$$\dot{\eta}(t) = -\frac{V_g(t)\cos\eta(t) - V_t(t)\cos[\psi_t(t) - (\psi(t) - \eta(t))]}{\rho(t)} + \dot{\psi}(t) \quad \eta(0) = \eta_0,$$

$$\dot{p}(t) = -\begin{bmatrix} V_g(t)\sin\psi(t) \\ V_g(t)\cos\psi(t) \\ V_{UAV}(t)\sin\gamma(t) \end{bmatrix} + \begin{bmatrix} V_t(t)\sin\psi_t(t) \\ V_t(t)\cos\psi_t(t) \\ V_h(t) \end{bmatrix}, \tag{27}$$

where

$$\beta_1(\omega(t)) = sign(\rho_s(t))\sqrt{\rho_s^2(t) + \rho_c^2(t)}, \qquad \beta_2(\omega(t)) = \tan^{-1}\left(\frac{\rho_c(t)}{\rho_s(t)}\right),$$
$$\rho_s(t) = -V_g(t) + V_t(t)\cos(\psi_t(t) - \psi(t)), \qquad \rho_c(t) = V_t(t)\sin(\psi_t(t) - \psi(t)),$$
$$V_t(t) = \sqrt{\omega_1^2(t) + \omega_2^2(t)}, \qquad \psi_t(t) = \tan^{-1}\left(\frac{\omega_1(t)}{\omega_2(t)}\right)$$

It is worth mentioning that in Equation (27), both the velocity of the UAV and the target are explicitly considered.

The relative position $p(t)$ can be calculated as:

$$p(t) = \begin{bmatrix} p_x(t) \\ p_y(t) \\ p_z(t) \end{bmatrix} = {}_C^I R p_c(t) = \left(\frac{h(t)}{-u(t)\sin\theta(t) + v\sin\varphi(t)\cos\theta(t) + \cos\varphi(t)\cos\theta(t)}\right) {}_C^I R \begin{bmatrix} u(t) \\ v(t) \\ 1 \end{bmatrix}$$

where

$u(t)\,\&\,v(t) =$ coordinates of target center in the camera frame.

${}_C^I R =$ coordinate transformation from the camera frame to the inertial frame.

$\varphi(t)\,\&\,\theta(t) =$ UAV’s known roll and pitch Euler angles for the rotation matrix ${}_C^I R$.

The control objective is to regulate $\rho(t)$ to $\rho_d$ where $\rho_d$ is a given desired 2D horizontal range between the UAV and the target. Furthermore, the relative altitude h(t) is not considered and Equation (27) reduces to

$$\begin{bmatrix} \dot{p}_x(t) \\ \dot{p}_y(t) \end{bmatrix} = -V_g(t)\begin{bmatrix} \sin\psi(t) \\ \cos\psi(t) \end{bmatrix} + V_t(t)\begin{bmatrix} \sin\psi_t(t) \\ \cos\psi_t(t) \end{bmatrix}$$

Let $$\omega(t) = V_t(t)\begin{bmatrix} \sin\psi_t(t) \\ \cos\psi_t(t) \end{bmatrix}, \qquad \omega(0) = \omega_0$$

The estimates of target's velocity $\hat{V}_t(t)$ and heading angle $\hat{\psi}_t(t)$ can be obtained through the following steps:

- State Predictor:

$$\dot{\hat{x}}(t) = A_m\tilde{x}(t) - V_g(t)\begin{bmatrix} \sin\psi(t) \\ \cos\psi(t) \end{bmatrix} + \hat{\omega}(t), \quad \tilde{x}(t) = \hat{x}(t) - x(t), \quad \hat{x}(0) = x_{0,}$$

- Adaptive Law:

$$\dot{\hat{\omega}}(t) = \Gamma_c \operatorname{Pr}oj(\hat{\omega}(t) - P\tilde{x}(t)), \quad \hat{\omega}(0) = \hat{\omega}_0$$

- Low–Pass Filter: Let

$$\begin{aligned} \omega_r(s) &= C(s)\omega(s), \quad \omega_r(0) = \hat{\omega}_0, \\ \omega_e(s) &= C(s)\hat{\omega}(s), \quad \omega_e(0) = \hat{\omega}_0, \end{aligned}$$

- Extraction of $\hat{V}_t(t)$ and $\hat{\psi}_t(t)$ from $\omega_e(t)$

$$\hat{V}_t(t) = \sqrt{\omega_{e1}^2(t) + \omega_{e2}^2(t)}, \quad \hat{\psi}_t(t) = \tan^{-1}\left(\frac{\omega_{e1}(t)}{\omega_{e2}(t)}\right)$$

More details on the developments of performance bounds, theorems and region of attraction can be found in [6]. The SUAV SIMULINK model used in this new adaptive guidance law is shown in Figure 21. The SUAV lateral dynamics

model used in this new guidance law is simple (with a single integrator function of the yaw rate command) and does not account for real UAV dynamics and processing delays.

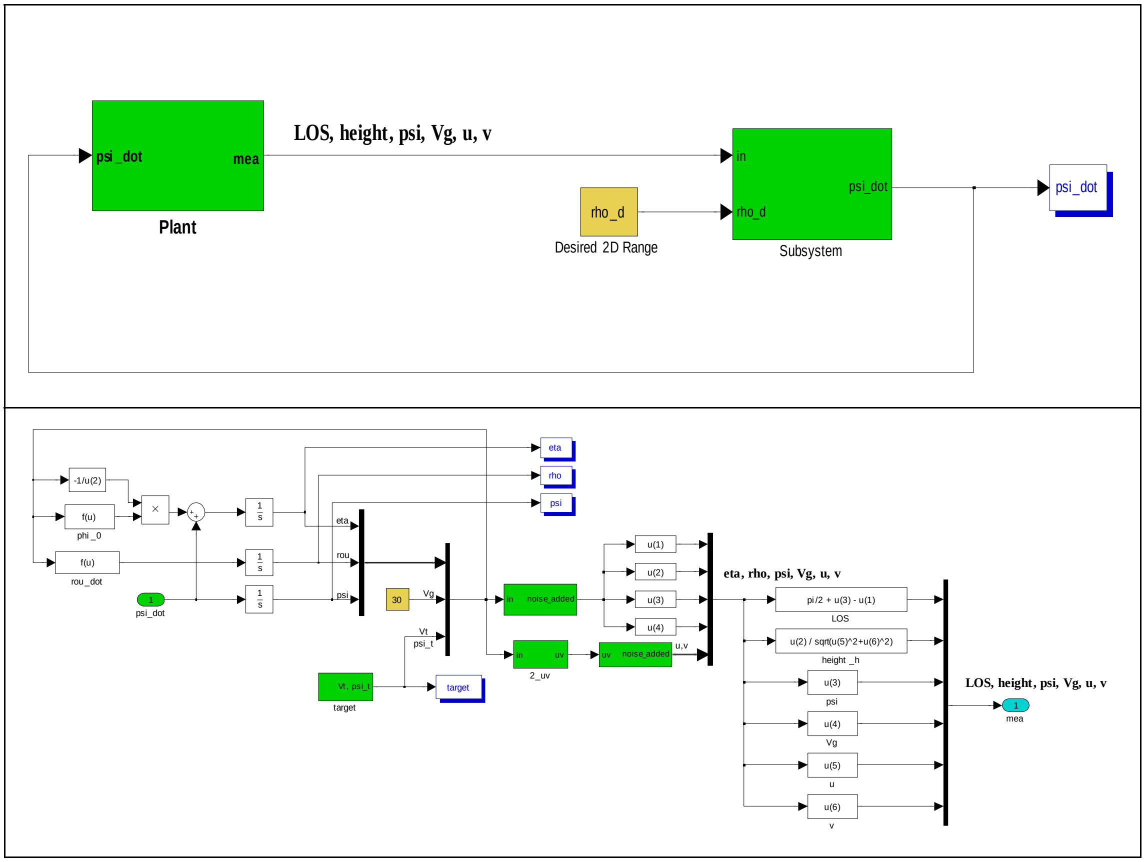



Figure 21. SUAV Model for New Adaptive Guidance Law

The 6-DOF SUAV model used in the current study accounts for environmental conditions and the 6-DOF dynamics of the SUAV. Dynamics of the 6-DOF SUAV model is non-linear and its precise numerical simulation is time consuming. Hence linearization of this model has been performed before integrating with the new adaptive control law. Linearization is performed using the MATLAB linear analysis tool in an open loop system at the following nominal conditions ($\dot{\psi}$=0, $\dot{\theta}$=0). The SUAV yaw rate $\dot{\psi}$ is the input point and $\psi$ is the output point resulting from the linearization $\psi$ = f($\dot{\psi}$). The entire loop 6-DOF SUAV model with an AP in feedback is illustrated in Figure 22.

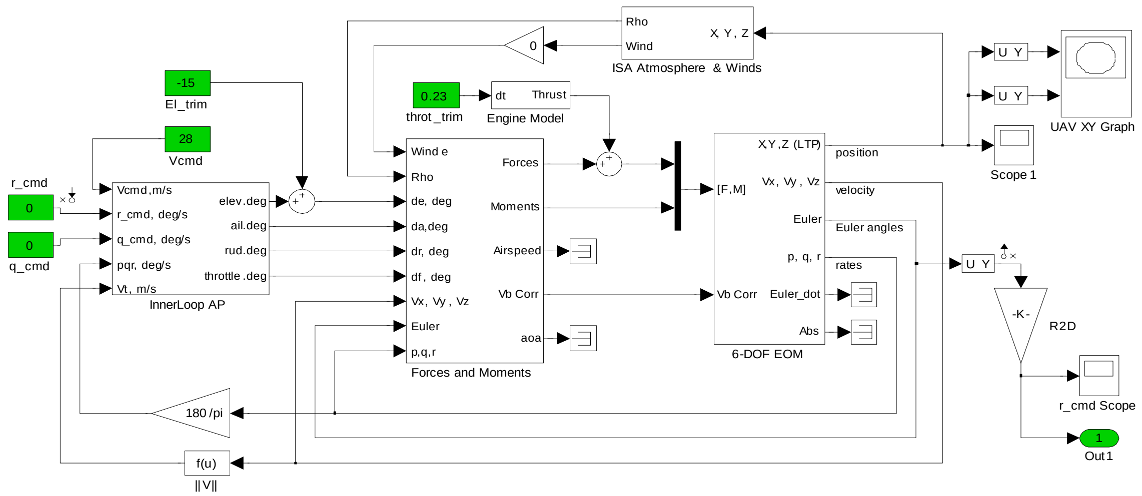



Figure 22. 6-DOF SUAV Model

The linearization results in a 10th order transfer function as shown:

$$\frac{-1.421e\text{-}013\,s^{10} - 2.183e\text{-}011\,s^9 - 1.397e\text{-}009\,s^8 + 11.63\,s^7 + 1445\,s^6 + 5.74e004\,s^5 + 7.139e005\,s^4 + 9.999e004\,s^3 - 3.497e006\,s^2 + 8.513e004\,s + 5.282e006}{s^{11} + 215.8\,s^{10} + 1.825e004\,s^9 + 7.552e005\,s^8 + 1.556e007\,s^7 + 1.484e008\,s^6 + 7.174e008\,s^5 + 2.147e009\,s^4 + 3.852e009\,s^3 + 3.205e009\,s^2 + 5.471e008\,s}$$

Reduction of the high order transfer function is beneficial as it eliminates the unobservable states and simplifies numerical calculation. This is done by using the ***minreal*** function in MATLAB. The resulting reduced order (7th order) transfer function is as follows:

$$\frac{7.105e\text{-}014\,s^7 - 7.294e\text{-}010\,s^6 - 5.988e\text{-}009\,s^5 + 11.63\,s^4 + 259\,s^3 - 422.1\,s^2 - 553.4\,s + 1056}{s^8 + 113.8\,s^7 + 3939\,s^6 + 4.1e004\,s^5 + 1.724e005\,s^4 + 4.54e005\,s^3 + 5.82e005\,s^2 + 1.094e005\,s - 3.024e\text{-}011}$$

A step response is used to compare the original linearized model and the reduced order model as illustrated in Figure 23. It is observed that the two models are fairly similar in response to a step command and thus it can be concluded that the reduced order model can also be used to represent the linearized 6-DOF SUAV model. From the plots, it is observed that the reduced order linearized model can be even further simplified while preserving the inherent dynamics. This simplified transfer function comprising of a time delay, T,

(in the form of $e^{-Ts}$) [7] and an integrator $\left(\frac{k}{s}\right)$ function is plotted for comparison and is observed to have a similar step response to the reduced order model. K is the gain of the integrator and determines the slope of the step response. From Figure 22, the time delay $\tau$ is estimated to be about 1 sec and forms the upper bound for the time delay.

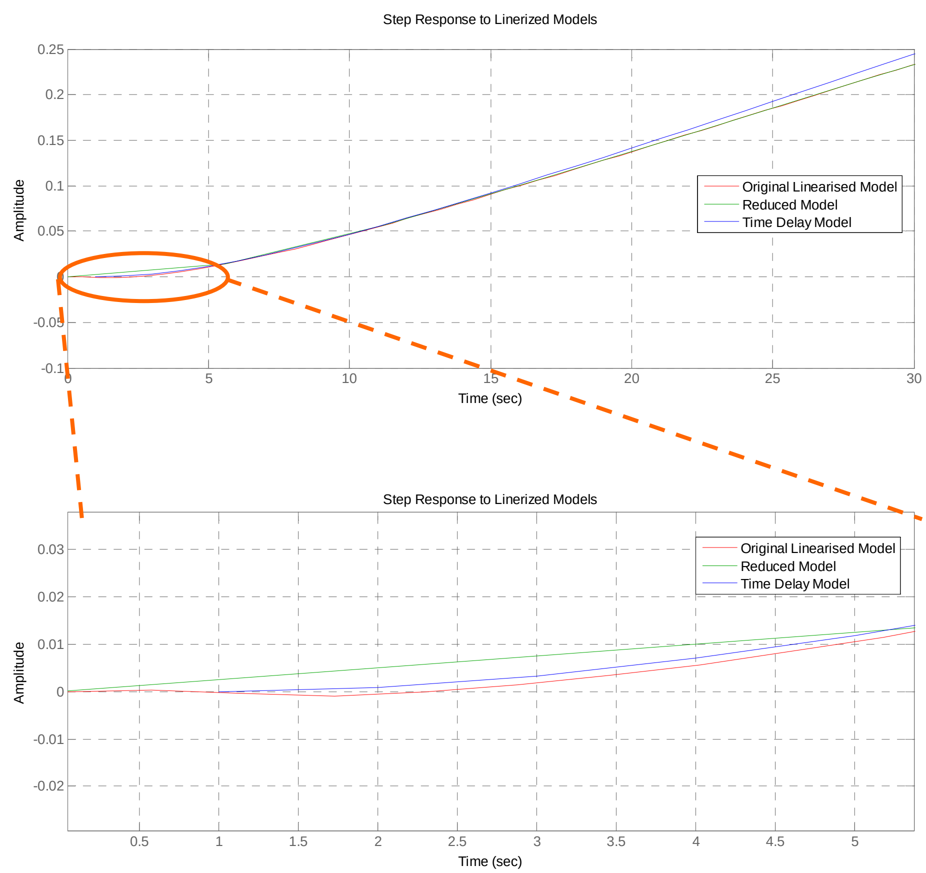



Figure 23. Step Response Comparison between Linearized Models

This time delay and integrator function is implemented into the new adaptive guidance SIMULINK model as shown in Figure 24.

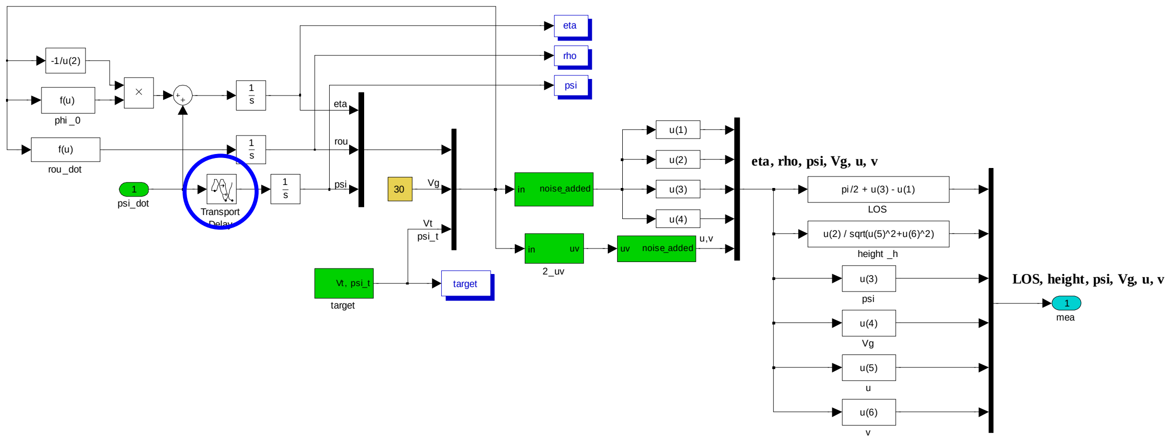



Figure 24. Implementation of Linearized 6-DOF SUAV Model

The simplified linearized model comprising of a time delay, T and an integrator $\left(\frac{1}{s}\right)$ function is subsequently tested for its "correctness" by comparing the SUAV tracking results for different values of time delay as seen in Figure 25 – 28. From the plots, it is observed that the as the time delay increases, the error in target speed and heading increases. Hence, the time delay model is only able to track the target effectively up to a limit of 0.5 sec time delay.

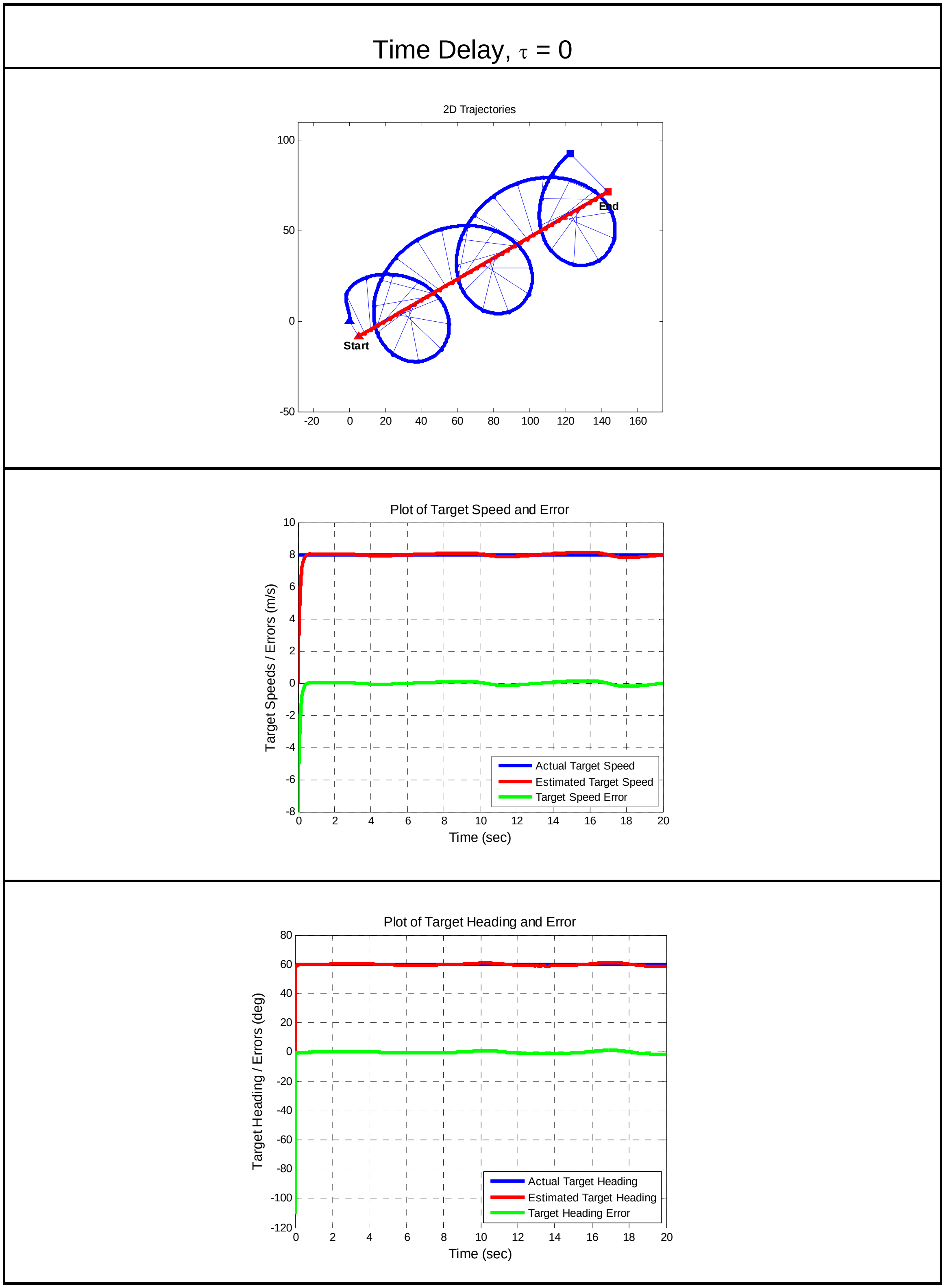



Figure 25. Effect on Target Tracking with Time Delay = 0

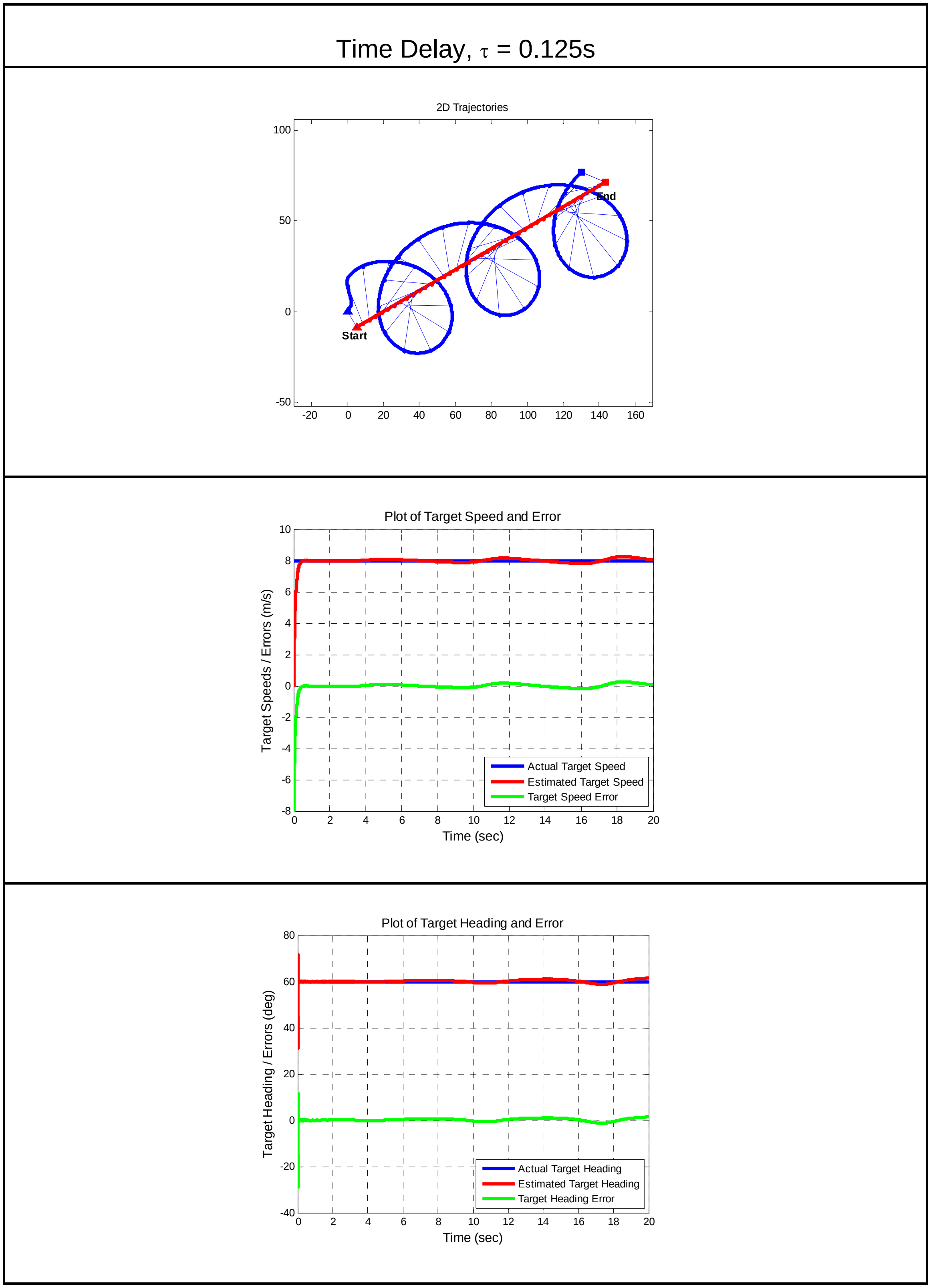



Figure 26. Effect on Target Tracking with Time Delay = 0.125s

Time Delay, $\tau$ = 0.25s

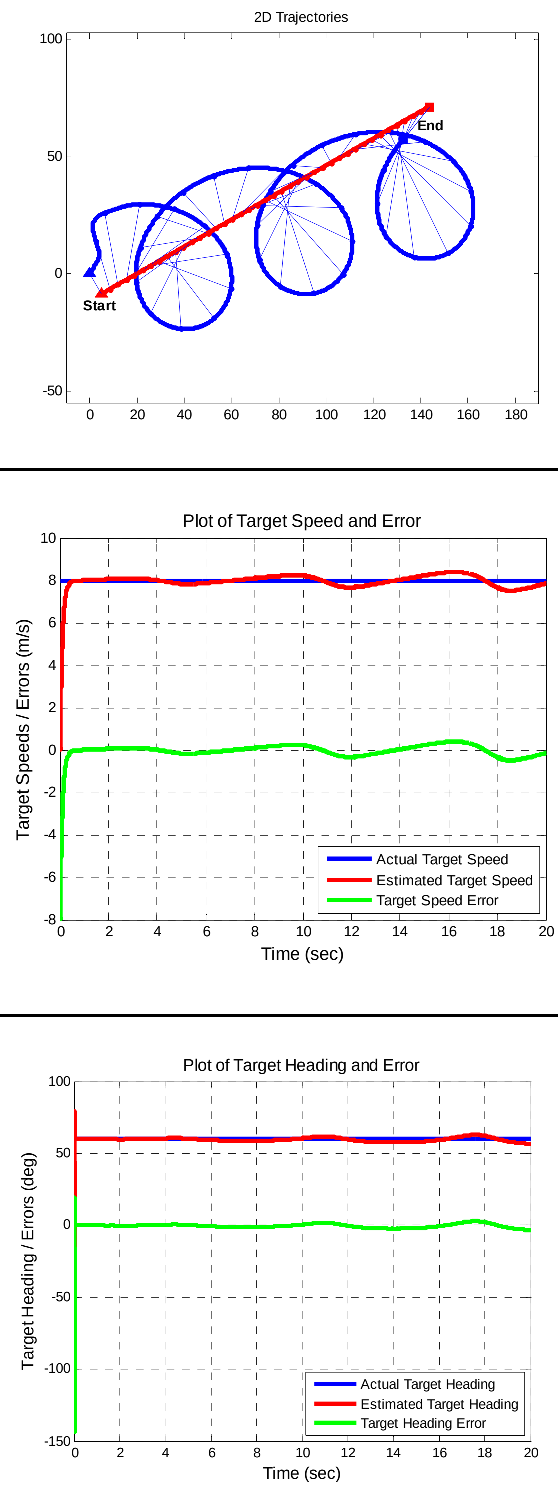



Figure 27. Effect on Target Tracking with Time Delay = 0.25s

Time Delay, $\tau$ = 0.50s

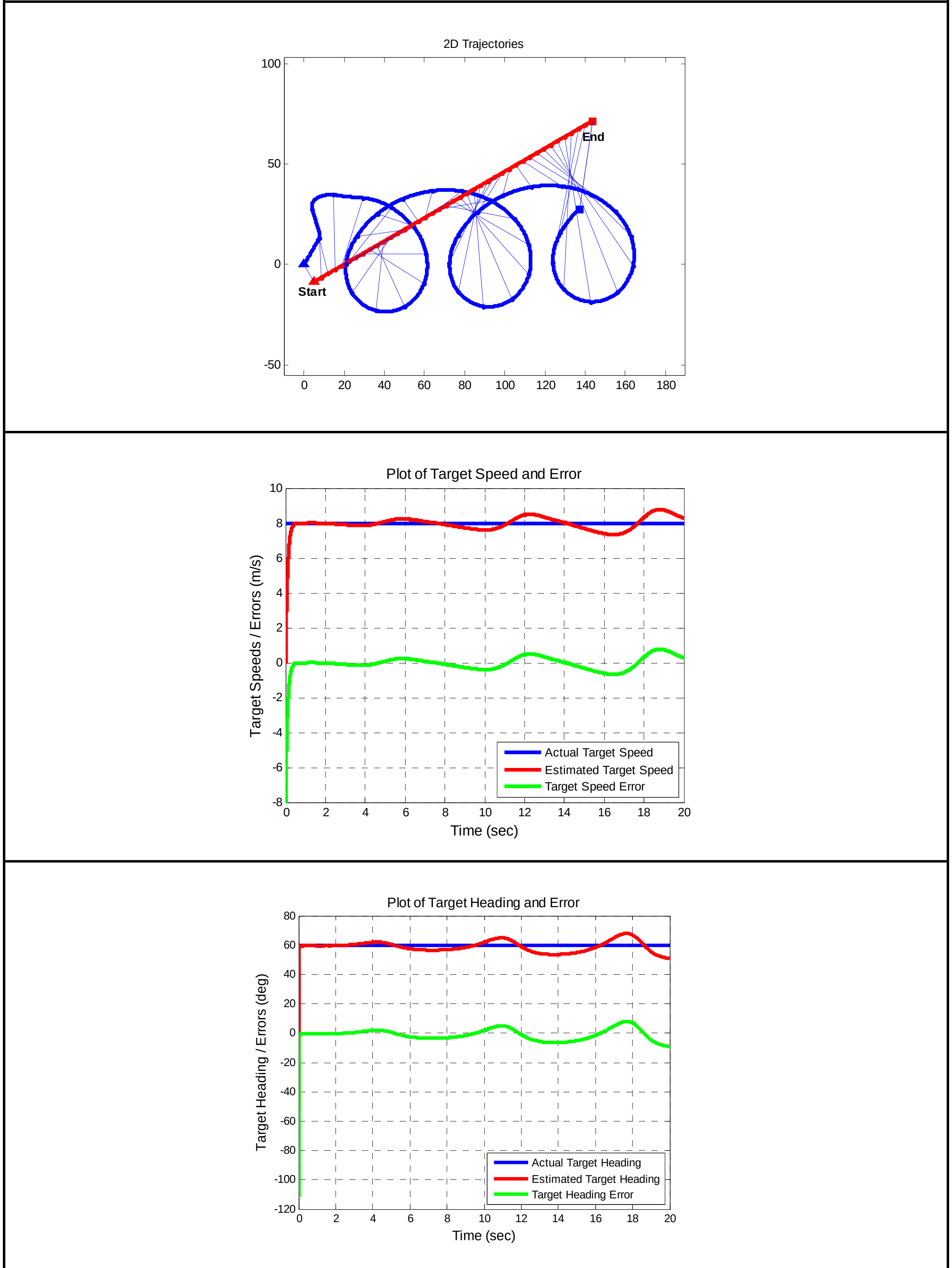



Figure 28. Effect on Target Tracking with Time Delay = 0.50s

THIS PAGE INTENTIONALLY LEFT BLANK

# IV. HARDWARE-IN-THE-LOOP SIMULATION

A cornerstone of the development environment is a hardware-in-the-loop (HIL) simulator. After testing the target escape scenarios in SIMULINK development environment, the next logical step is to implement the control law in the hardware-in-the-loop simulation before the actual flight test.

The simulator allows the aircraft control laws and mission functionality to be tested without risking the hardware in flight test. It also provides an ideal training tool that can be used in the lab. Although HIL simulation cannot replace flight testing, it measurably reduces the likelihood of failure by detecting bugs and deficiencies before the hardware is put at risk. The schematic of the hardware-in-the-loop setup is shown in Figure 29. The different components of the hardware setup are described in this section.

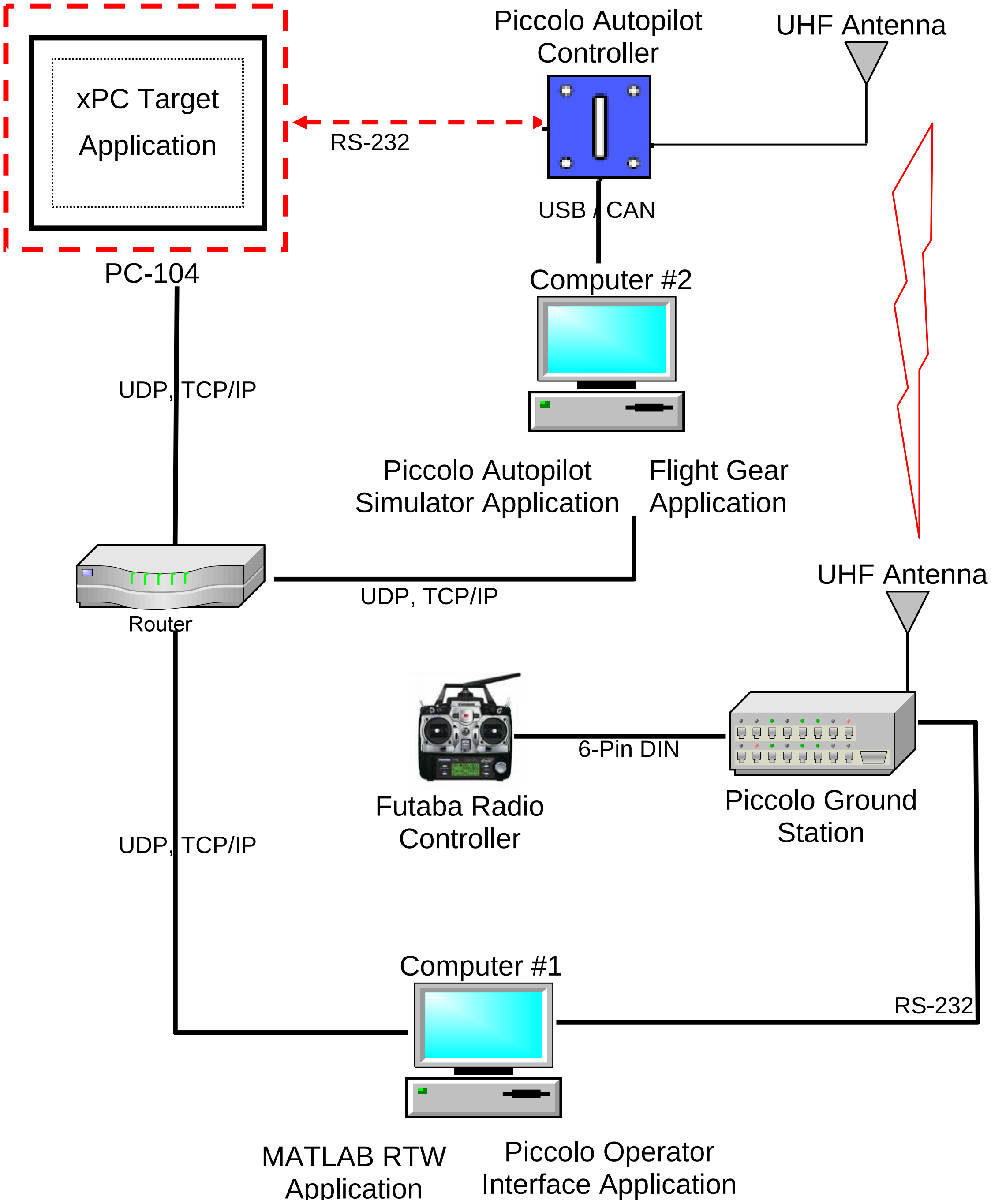



Figure 29. Schematic of Hardware-in-the-Loop Setup

## A. PC/104 DESCRIPTION AND SETUP

The PC/104 system is a popular standardized form-factor platform for small computing modules. They are typically used in industrial control systems or vehicles with embedded control applications. As shown in Figure 30, it comprised of ultra-compact (3.6" x 3.8") stackable modules. Each stack must contain at least one motherboard (CPU), which acts as a controller for the peripheral components. The motherboard is often referred to as a single board computer (SBC), for it often has interfaces for all standard PC components (e.g., keyboard, mouse, serial ports, ethernet port etc.). It should be noted that the components heat up quickly and adequate cooling means should be provided to prevent overheating.

Figure 30. PC/104 System

The PC/104 [8] is the “brain” of the hardware setup. It executes the xPC target model uploaded from the Host PC. MATLAB / Real Time Workshop must be correctly configured according to the step-by-step instructions detailed in [9] before compiling/building the xPC target model in PC/104. The xPC target model is uploaded from the Host PC to PC/104 via a TCP/IP network. The communication link between the PC/104 and the Piccolo AP controller is through a RS232-CAN bus cable for data exchange. For further data analysis and real-time graph plotting, the xPC target model outputs can be similarly extracted via UDP/IP at near real-time to another SIMULINK data logging model.

## B. PAN-TILT UNIT DESCRIPTION AND SETUP

The PTU-D300 Pan-Tilt Unit as seen in Figure 31 from Directed Perception [10] provides fast, accurate, and durable positioning of cameras, antennas, lasers, and other large payloads of up to 70 lbs. The built-in command set offers both ASCII and binary formats. This command set supports real-time control at up to 60 commands per second with very low and predictable latencies.

Figure 31. PTU-D300 from Directed Perceptions

In the hardware setup, the PTU-D300 models the motion of the UAV. This is achieved by converting the Euler angles (roll, pitch and yaw) obtained from Piccolo autopilot into azimuth and elevation via a series of coordinate frame transformation. The PTU-D300 can be connected to a host computer via the “Hyper Terminal” user interface available in Microsoft® Windows operating system for the purposes of parameter configuration. In addition, the unit can also be connected to the PC/104 system using a standard RS-232 serial cable during simulation runs. It is essential to consider the mechanical pan-tilt limits and speeds of the device and ensure that they are not exceeded during simulation runs to prevent saturation of the output signal. In addition, the operating speeds

in the pan-tilt direction should also be correctly configured to ensure desired tracking response. The limits of the device are detailed in Table 1.

| **Resolution** | **Sec Arc/Position** | **Deg/Position** |
|---|---|---|
| | 92.5714 | 0.025714278 |
| **Position Properties** | **Positions** | **Degrees** |
| **Minimum Pan Position** | -6803 | -174.93 |
| **Maximum Pan Position** | 6804 | 174.96 |
| **Minimum Tilt Position** | -3516 | -90.41 |
| **Maximum Tilt Position** | 1169 | 30.06 |
| **Speed Properties** | **Positions/Sec** | **Degrees/Sec** |
| **Minimum Pan Position** | 1985 | 51.04 |
| **Maximum Pan Position** | 0 | 0.00 |
| **Minimum Tilt Position** | 1985 | 51.04 |
| **Maximum Tilt Position** | 0 | 0.00 |
| **Acceleration Properties** | **Positions/Sec$^2$** | **Degrees/Sec$^2$** |
| **Minimum Pan Position** | 2000 | 51.43 |
| **Maximum Pan Position** | 2000 | 51.43 |

Table 1. Operating Limits of the PTU-D300

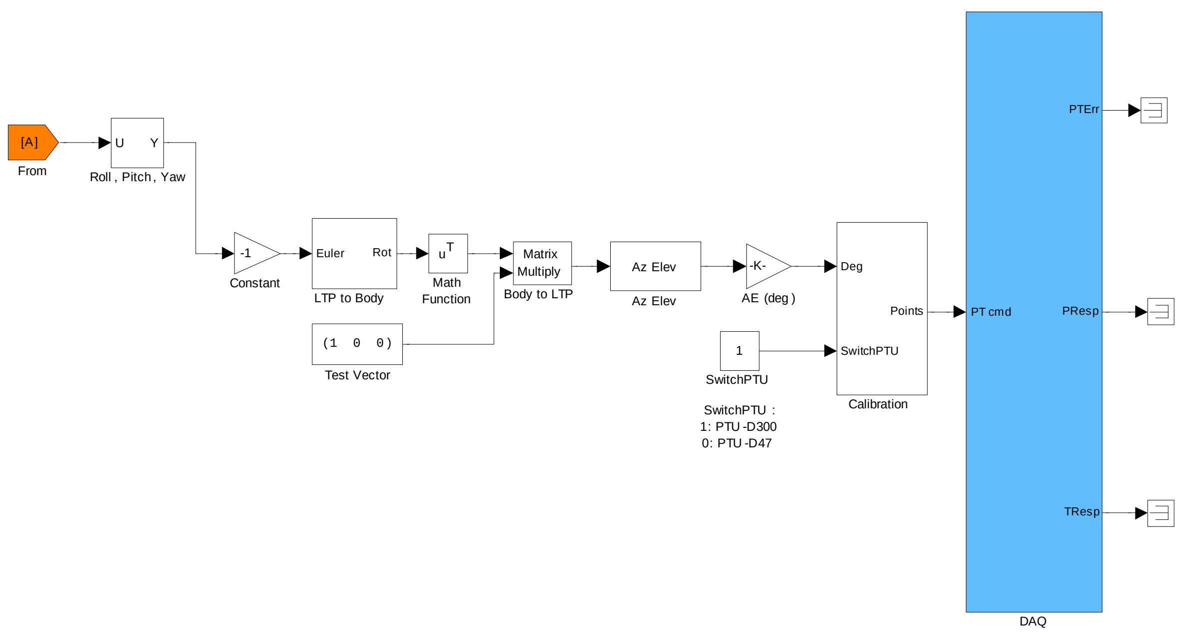



Figure 32. xPC Driver Model for the PTU-D300

Figure 32 shows the xPC target model that drives the PTU-D300. The model first obtains the three Euler angles measured by Piccolo AP. Through a series of frame rotation and transformation, these three Euler angles are subsequently converted into azimuth (pan) and elevation (tilt) commands which drives the motion of the PTU.

## C. PICCOLO PLUS AUTOPILOT OVERVIEW AND SETUP FAMILIARIZATION

Piccolo Autopilot is a complete integrated avionics system from Cloud Cap Technology for small unmanned aircraft. There are four versions of Piccolo available in the market and the one used in the hardware in the loop setup is a *Piccolo Plus.* The Piccolo control system consists of four main parts: an avionics control system (mounted onboard the SUAV during flight tests), a ground station, a computer for the operator interface application and mission monitor, and a pilot manual control interface via a modified Futaba radio controller.

The Piccolo system employs two separate control loops: the faster inner loop controls the aircraft dynamics within the aircraft itself and the slower outer loop controls the path that the aircraft is expected to follow via a wireless communication link between the piccolo autopilot controller and the ground control station. The control law that has been developed will be utilizing the inner control loop to realize the flight control of the SUAV. Typical Piccolo HIL equipment and its setup are shown as follow in Figure 33. [11], [12].

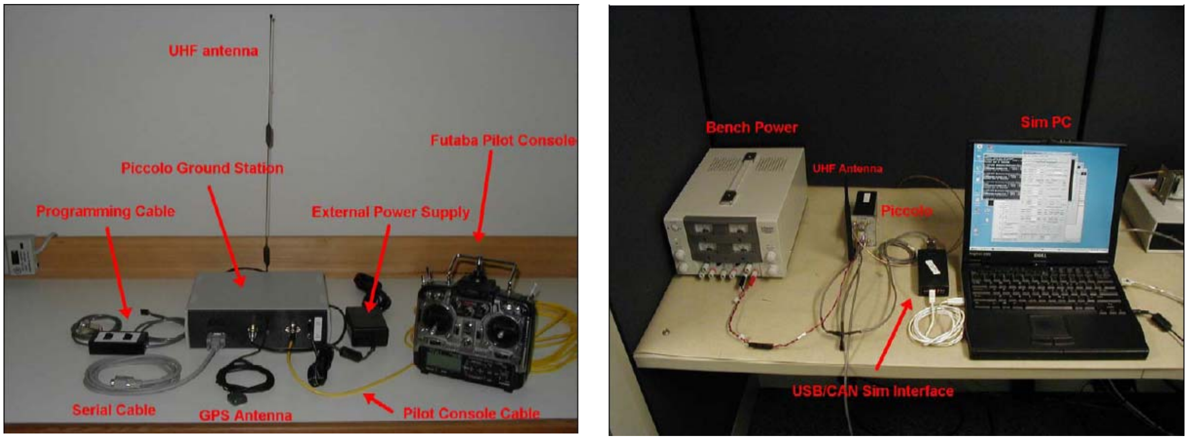



Figure 33. Ground Station Setup (without Operator Interface PC), SIM PC and Avionics Side of HIL Simulation

### 1. Avionics – Piccolo Autopilot (AP) Controller

The CPU of Piccolo AP is the MPC555 microcontroller, which is a new breed of automotive controller based on the PowerPC architecture, capable of delivering 40 MHz PowerPC operation, including hardware floating point.

Integrated within the avionic controller unit are three ADXRS300 gyros and two two-axis ADXL210e accelerometers. The Motorola M12 GPS provides Piccolo with its basic groundspeed and position. Included with the Piccolo interface are a dual ported mpxv50045 4kPa dynamic pressure sensor, an absolute ported mpx4115a barometric pressure sensor, and a board temperature sensor. A sophisticated data link that is built on the MHX 910/2400 radio modem from Microhard Systems Inc. provides up to 40Kbaud of throughput and is used for command and control, autopilot telemetry, payload data transfer functions and differential GPS corrections. The frontal panel included the filtered 44-pin vehicle interface connector, GPS and UHF antenna SMA connectors as well as the Pitot and Static pressure port nipples. The Piccolo block diagram and front panel schematic is shown in Figure 34.

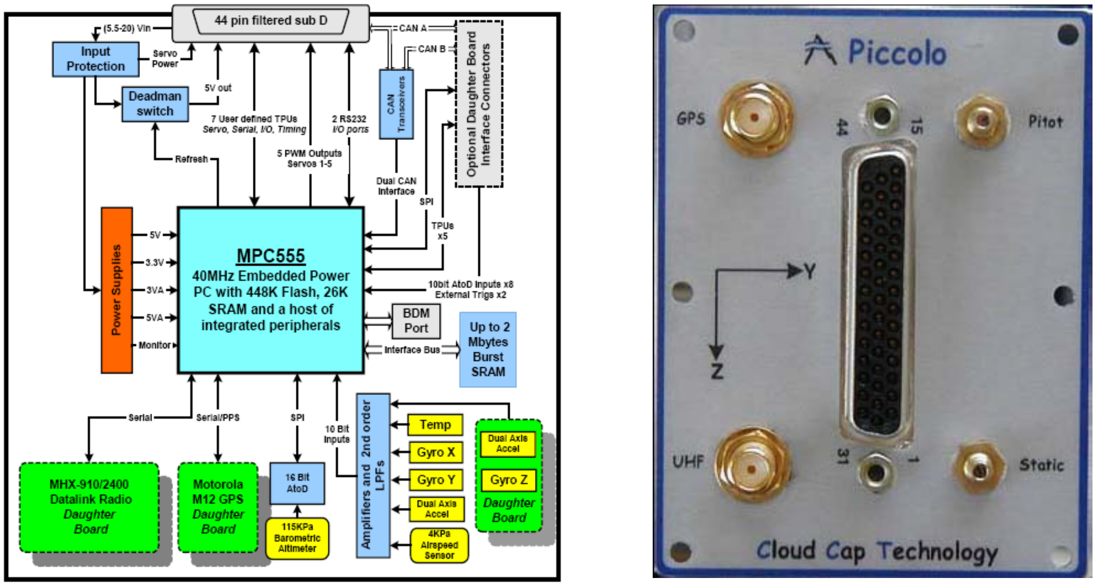



Figure 34. Piccolo Block Diagram and Front Panel

In the HIL setup and simulation environment, the Piccolo AP is connected to Computer #2 which runs the flight dynamics Simulator software. The inter-link between Computer #2 and the Piccolo AP controller is via a USB-CAN (computer area network) data exchange cable through the CAN port. In addition, Piccolo AP is linked to PC/104 via a RS232 serial cable through the program port. Data exchange is done via COM1, running at baud rate 57,600.

The Simulator allows the aircraft control laws and mission functionality to be tested before the actual flight testing. It reduces the likelihood of failure by detecting bugs and deficiencies before the aircraft and related hardware are put at risk.

After launching the Simulator program, the program is initialized in two simple steps. First, under the FILE menu options, a default "cub aircraft" model file is selected as its flight characteristics resemble that of the actual Tele-Master SUAV that is in used for flight testing. Secondly, the start state of the Simulator also has to be initialized. A default file containing the geodetic data around the Camp Robert Airfield test site is available in the system for quick initialization.

The key parameters of interest are the flying altitude of 300m and a flying speed (TAS) of 28m/s. The “Reset” “Apply slew” and “clear slew” radio button apply the typed customized parameters to the Simulator program. The Simulator program is started and stopped using the “Start” and “Stop” radio button. Turbulence parameters can be input to the program to simulate weather and wind conditions. The Simulator program interface layout is shown in Figure 35.

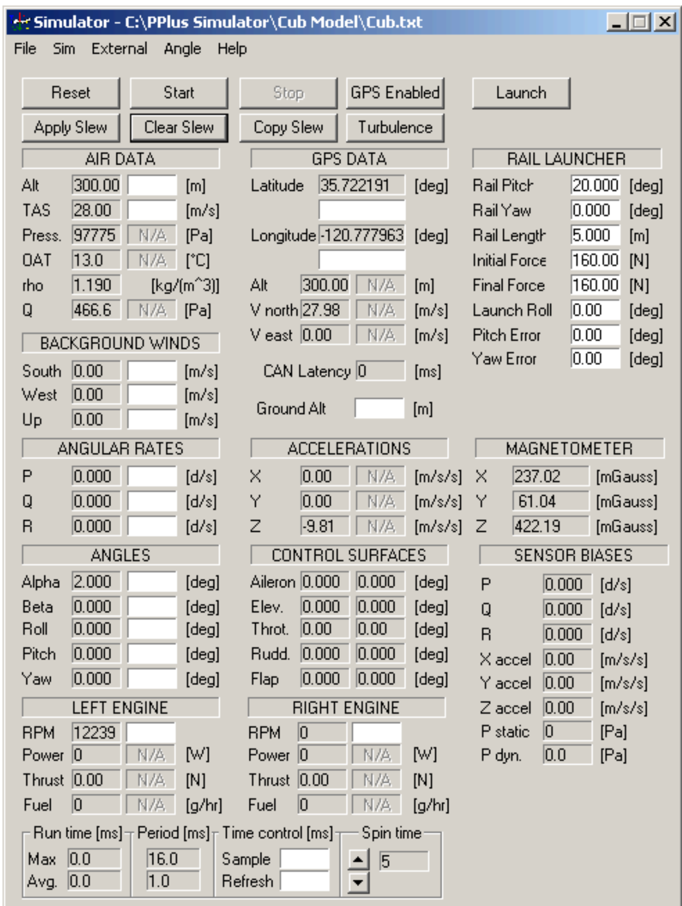



Figure 35. Simulator Program Interface

### 2. Piccolo Ground Station

Piccolo’s ground control station is based upon the same hardware that makes up the avionics package. It manages the communication link to one or more avionics systems, interfaces to the pilot in the loop console, and provides a

command and control stream to the operator interface PC. The Ground Station connects to the operator interface PC through a standard 9-pin serial cable. The ground station GPS Antenna is connected to the rear panel SMB connector and the ground station UHF antenna is hooked up to the BNC connector. A 6-pin circular DIN pilot console cable connects the pilot in the loop command to the Ground Station through a modified Futaba radio controller. The Ground Station exchanges data with the Piccolo avionics via a built-in 900 MHz or 2.4 GHz ISM band radio wireless link made from the MHX-910/2400 frequency hopping radio from Microhard Systems Inc. Piccolo Ground Station is shown in Figure 36.

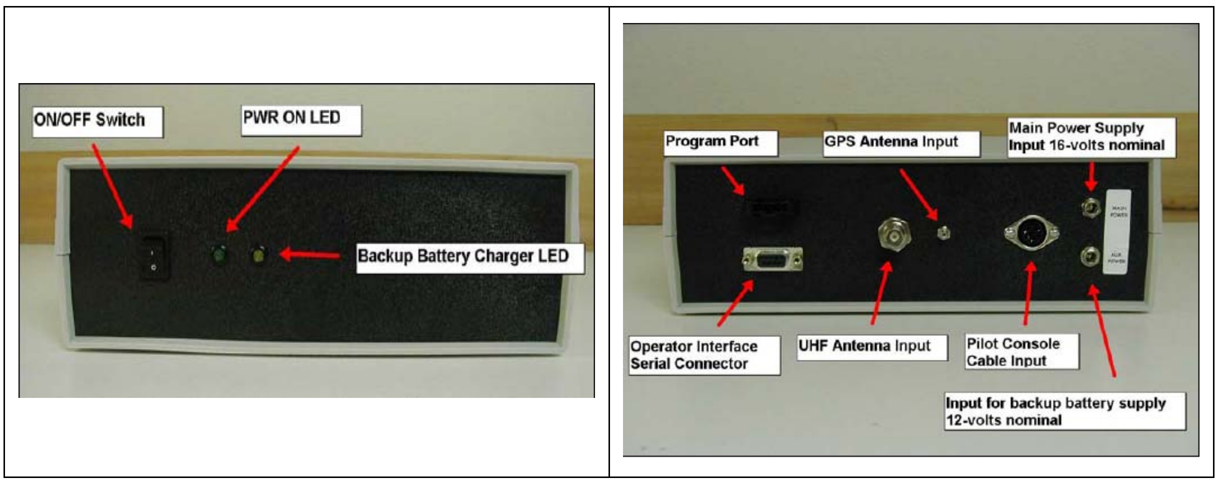



Figure 36. Piccolo Ground Station Front and Back Panels

The operator interface (OI) is a software system that runs on a Windows PC and most of the system features are accessible from this interface. It provides a command and control interface for Piccolo operators. The operator interface communicates to the Ground Station over a RS-232 serial link (default to COM1). Installation of the OI is through a Windows installer file, “Operator Interface.msi” which is downloadable from the Cloud Cap Technology site. The OI is installed in Computer #1.

The OI provides two station screens for operator interface with the Ground Station. The Ground Station screen provides a Window Menu and Unit Menu for

user to manage the avionic window display layout and units of telemetry to be displayed. The ID number of the Piccolo AP will have to be added under the "Network Control" interface. In addition, the power ratings under the "UHF Radio Settings" interface have to be adjusted to about 0.001W to minimize radiation during laboratory testing. An advance option under the Window Menu displayed more advanced version of the avionics window for operator interface. The avionics window displayed the telemetry data received from Piccolo avionics. A screenshot of the Piccolo OI is displayed in Figure 37.

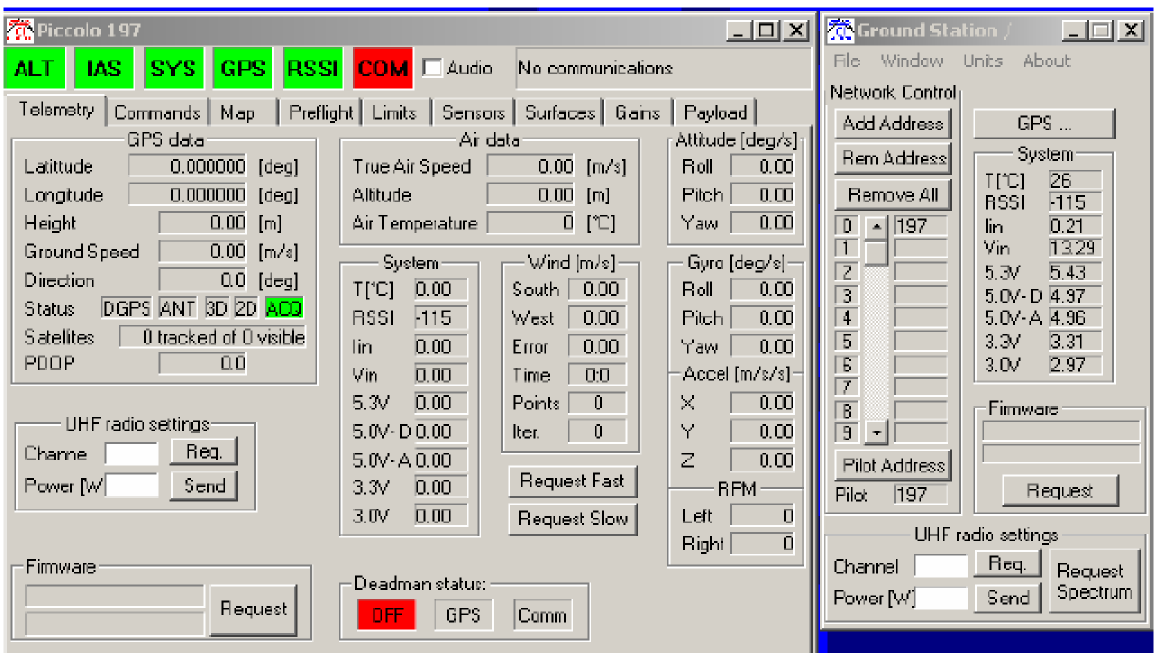



Figure 37. Screenshot of Piccolo Operator Interface

The essential telemetry information is displayed in tabulated pages which are user selectable. The tabulated pages are arranged in the following order: (1) Telemetry, (2) Commands, (3) Map, (4) Preflight, (5) Limits, (6) Sensors, (7) Surfaces, (8) Gains, (9) Payload and (10) Parameters. Some essential pages that are required in HIL setup will be briefly described in subsequent paragraphs.

### *a. Telemetry Page*

The telemetry page displays data from all the sensors of the system. Sensor information that are displayed includes GPS data, Air Data, Sonic altimeter, MHX radio settings, System Version, Diagnostic, Wind, Attitude, Gyro, RPM and Compass. The request fast and request slow buttons alter the rate at which the telemetry data are sent. Default slow mode sends data once per second; fast mode sends data 20 times a second or limited by available bandwidth.

### *b. AP Commands*

The autopilot commands page displays the current autopilot command status, and allows the user to change IAS, altitude, turn rate, flaps and waypoint tracker via an interactive input interface. Other commands like autopilot mode, stick mode, engine control and flight actions are selectable through on-screen radio buttons.

### *c. MAP*

The map page displays the current location of the vehicle, and provides an interface for creating flight plans. The map is built on ESRI's MapObjects, which means that it is capable of displaying geo-referenced raster files, as well as vector shape files. Customized map in TIF format can be added in using the add image layer or add vector layer command.

### *d. AP Limits*

The limits page is used to view and alter the autopilot and mission limits. Autopilot limits include min/max of airspeed, altitude, bank angle, aileron, elevator, rudder, throttle and flap and can be input via interactive interface. A request and send limit radio button is used to interrogate and alter the current limit on the Piccolo avionics. Detailed procedures on setting of AP limits are found at Cloud Cap Technology web site:

(http://www.cloudcaptech.com/download/Piccolo/Version%201.3.2/Docs/Piccolo% -Page 41, 4.4.3.1)

### *e. Sensors*

The sensors page gives the current sensor readings and calibration information for each sensor. It is normally only used during the setup process. An important step in the HIL setup is to check and align the Euler angle axis of Piccolo avionics with respect to the aircraft body frame out of twenty-four different possible axes orientation. A sample page is shown in Figure 38.

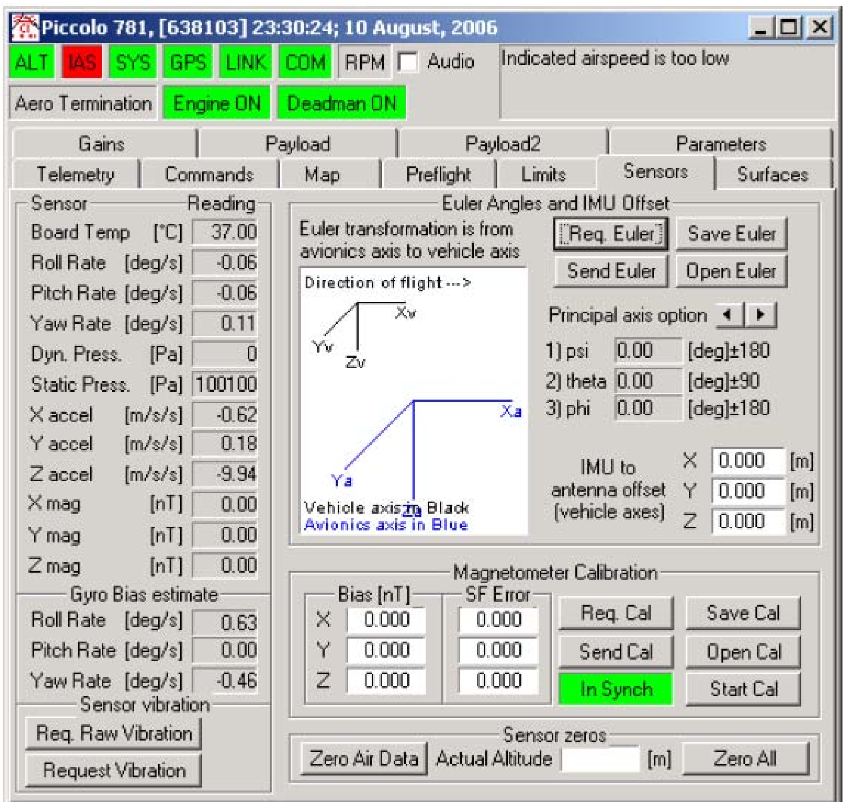



Figure 38. Alignment of Avionics Orientation to Aircraft Body Frame

### *f. AP Gains and Trims*

The gains page is used to view and alter the autopilot gains and trims. There are gains for eight loops. Before the Piccolo AP can be gainfully utilized for HTL simulation and control, the gains of the AP aircraft model have to

be tuned correctly. Detailed procedures on tuning of AP gains and trims are found at Cloud Cap Technology web site:

(http://www.cloudcaptech.com/download/Piccolo/Version%201.3.2/Docs/Piccolo% (Page 41 – 43, 4.4.3.2/4.4.3.3)

### 3. Flight Gear Visualization Environment Familiarization

During HIL simulation, it is necessary to be able to visualize the effects that the control law have on the flight performance and stability of the aircraft. An open source application called Flight Gear is provided for the visualization of aircraft attitude through an UDP/IP network protocol. In order to get Flight Gear to accept the state packet it must started with the correct command line switches. A batch file "runflightgear.bat" is provided for this. The Flight Gear application is installed on Computer #2 equipped with a video card designed for OpenGL hardware acceleration. The flight visualization is activated from the Piccolo Simulator Flight Gear output interface selection (Computer #2). A screen shot display of the Flight Gear visualization is shown in Figure 39.


Figure 39. Screen Shot Display of Flight Gear Visualization

### 4. HIL Network Communication and Data Exchange

#### *a. Router Network Communication Protocol*

TCP/IP and UDP/IP are the two forms of network communication protocol used in the HIL network communication setup. TCP/IP is utilized for the uploading of xPC Target model from Host PC to PC/104. UDP/IP is utilized for the downloading of xPC Target model outputs from PC/104 to Flight Gear's computer for visualization of the aircraft attitude and performance, as well as extraction of relevant xPC Target data to another SIMULINK Data Extraction program running on Host PC (Computer #1). The xPC Target data are downloaded for post simulation data analysis and real-time graph plotting utility purposes.

UDP/IP is described as being connectionless and unreliable, the sending party will send out information whenever it can, and the receiver will receive information whenever it is able to do so. Information that is sent when the receiver is not available will simply be lost. UDP/IP is widely employed as the choice network communication protocol for real-time applications, since only the most recent information is of relevance.

#### *b. Setting Communication to Piccolo AP in HIL*

Serial Interface (SI) communication technology is widely employed in data interfacing in modern control in establishing the link between the ground control unit and the airborne vehicle. Mathworks' xPC target provides an RS232 library that supports a variety of serial instruments. These drivers support synchronous, asynchronous, and binary (asynchronous) communication modes and should be added to the MATLAB existing libraries before use on the control model. Although the latest xPC target release supports RS232/422/485 protocols and a number of serial communication boards including Quatech and Diamond's products, it still does not extend the reading capability of new serial data formats, such as proprietary communication protocol in Piccolo AP.

The application of a binary asynchronous communication that is supported directly by the standard Mathworks library relieves the user of having to develop an actual RS232 hardware driver. It conveniently allows focusing on the implementation of data interfacing rather than the hardware programming. Since the majority of SI formatting is proprietary, the most efficient way to implement the SI capability consists in writing Level-2 S-functions. The technique used in this thesis provides the operational separation of principal functions among the pre-built library blocks and the user-developed S-functions. Standard serial communication blocks deliver raw binary data to the model using the optimized Mathworks routines. The S-functions perform format specific processing that is not readily available. Figure 40 depicts this concept [5].

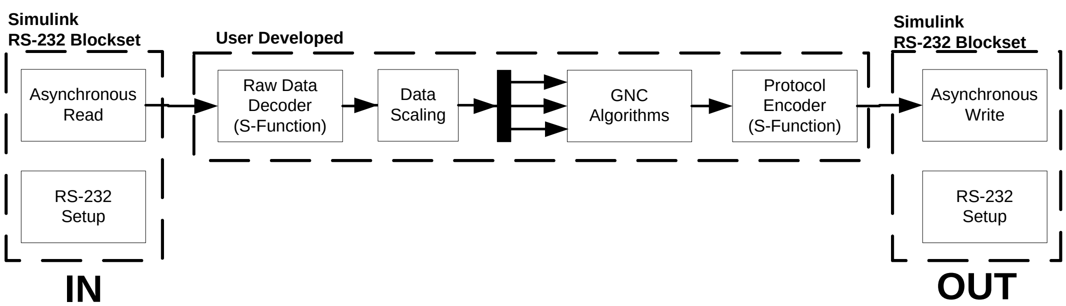



Figure 40. Separation of Interface Functions

A customized SIMULINK RTW communication program has been developed for the data exchange between xPC Target model and Piccolo avionics for use in the control law SIMULINK model as shown in Figure 41.

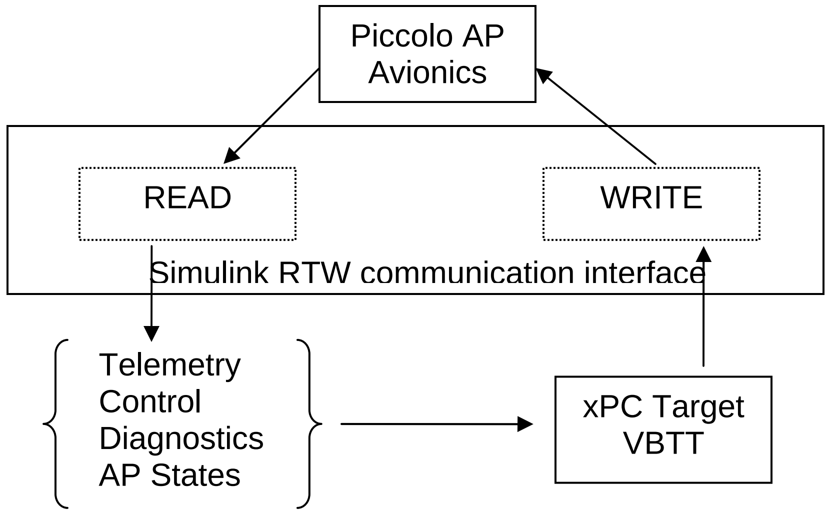



Figure 41. SIMULINK RTW Communication Interface

Using the customized SIMULINK RTW communication interface program, relevant telemetry, control, diagnostics and AP-states information can be readily extracted from Piccolo avionics output for the xPC Target model guidance control computation. The computed control commands from xPC Target model can be translated into Piccolo communication protocol format for control of Piccolo avionics.

The SIMULINK RTW communication interface program comprised of MATLAB standard RS232 serial binary communication blocks and customized programmed Level-2 S–functions. The SIMULINK RTW communication interface program applies color-coded blocks and paths to represent the WRITE (GREEN) and READ (RED) blocks and paths between xPC Target and Piccolo avionics. To ensure that the SIMULINK RTW communication block is communicating to the correct Piccolo avionics unit, the appropriate Piccolo avionics unit ID (indicated on the Piccolo unit) has to be input properly in the program.

### c. *Reading from Piccolo AP*

The READ block of the SIMULINK RTW communication interface has five customized S-function blocks namely: “pplus_readstream.c” “pplus_autopil.c” “pplus_diag.c” “pplus_control.c” and “pplus_telemetry.c” nested within. The input parameters for S-function “pplus_readstream.c” are Piccolo Autopilot ID (APID) and buffer size (BUF). The output from the S-function comprised of “data” and “header type” are parsed through a decoder sub-block. The other four S-functions that are nested within the decoder sub-block are utilized to decode the proprietary Piccolo communication protocol to separate out the Piccolo telemetry, control, diagnostics and AP states information. The READ block of the RTW communication interface program is depicted in Figure 42.

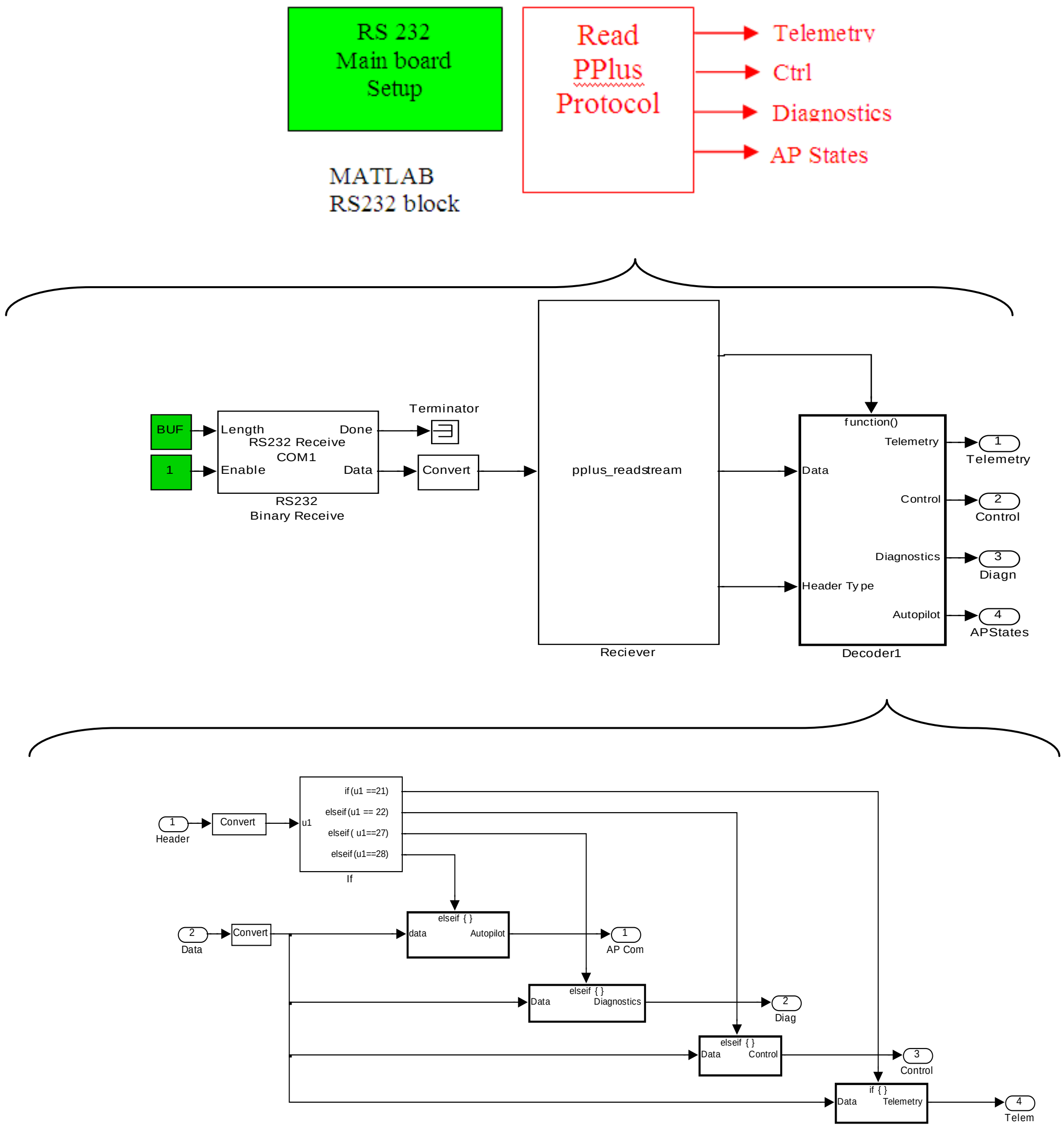



Figure 42. READ Block of SIMULINK RTW Communication Interface Program

### *d. Writing to Piccolo AP*

The WRITE block of the SIMULINK RTW communication interface program converts and codes the xPC VBTT guidance control message into proprietary Piccolo communication protocol before sending it through the standard MATLAB RS232 serial binary communication block.

A MATLAB sub-block is utilized to convert the xPC Target guidance commands into an “AP payload data stream”. The input to the sub-block is the AP control loop number, control type and control command values. The output from the sub-block is eight bytes of “AP loop message” also known as the “Payload” for the Piccolo AP.

An AP Stream Wrapper block is utilized to code the “AP loop message” or “Payload” into proprietary Piccolo communication binary data stream. The proprietary Piccolo communication binary data stream is coded as a 2-layers communication protocol format. An outer layer is utilized to differentiate the type of data stream for “header” and “checksum” purposes, and an inner layer is utilized to differentiate the type of command payload “packet” information. Three customized S-functions, namely “toplevelcrc.c” “enc-apilot_loop_fix.c” and “enc_top_level26.c” are nested within the WRITE block, to perform the checksum and proprietary coding functions. The coded binary data stream is subsequently transmitted through MATLAB standard RS232 serial communication block to Piccolo AP. A typical WRITE block of the RTW communication interface program is depicted in Figure 43.

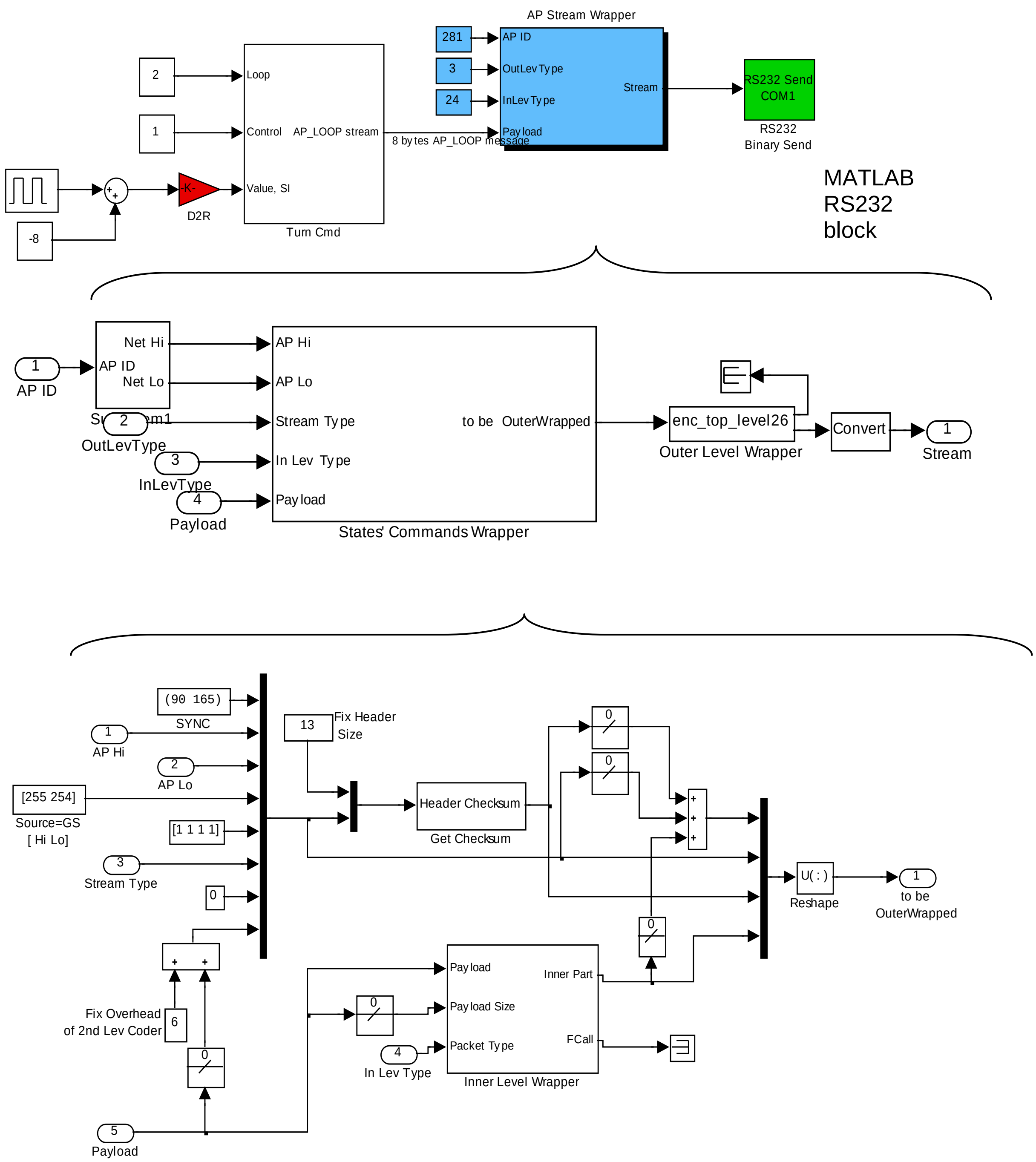



Figure 43. WRITE Block of SIMULINK RTW Communication Interface Program

A summary of the S-functions and their functions are tabulated in Table 2.

| Where | Name of S-Functions | Functions |
|---|---|---|
| READ Block | pplus_readstream.c | Extract messages from the Piccolo RAW data, mark output with a header |
| READ Block | pplus_autopil.c | Parse Piccolo AP states |
| READ Block | pplus_diag.c | Parse Piccolo diagnostic |
| READ Block | pplus_control.c | Parse Piccolo Control |
| READ Block | pplus_telemetry.c | Parse Piccolo Telemetry |
| WRITE Block | enc_top_level26.c | Outer Level Wrapper |
| WRITE Block | Toplevelcrc.c | Header / Checksum |
| WRITE Block | enc-apilot_loop_fix.c | Inner Level Wrapper |

Table 2. Summary of S-Functions

An “INIT-GUI” M-script file is utilized to initialize all the Piccolo AP state variable names and values that are used in the SIMULINK RTW communication interface program. The “INIT-GUI” M-script file is placed under the “Model Properties/Model callbacks” block for initialization.

A separate UDP communication SIMULINK program utilizing the “Pack, UDP-Send-Binary” in SIMULINK library is used to output the xPC Target output data stream (from PC/104) to another SIMULINK Data Extraction program running on Computer #1. The SIMULINK Data Extraction program utilizes the “UDP-Receive-Binary, UnPack” UDP communication block to receive the data stream. This arrangement is to alleviate the computational constraints of PC104 in displaying the real-time data stream. The extracted xPC Target data are

utilized for post simulation data analysis and real-time graph plotting utility for visualization of the simulation progress (on Computer #1).

## D. HARDWARE-IN-THE-LOOP (HIL) SIMULATION RESULTS

This section describes implementation of the developed control law in the HIL environment together with commercial-off-the-shelf Piccolo Plus AP and testing of the control law performance in the scenarios discussed previously as follows:

- **Scenario A:** Stationary target.
- **Scenario A1:** Target is stationary with different values of time delay.
- **Scenario C:** Target is moving with a nominal speed (8m/s) with variable heading.
- **Scenario C1:** Target is moving with a nominal speed (8m/s) with varying heading and different values of time delay.

It should be noted that testing in the HIL simulation environment is the last testing stage before implementing any control algorithms in real flight. Due to the fact that the inner-loop controller (autopilot) of the initial SIMULINK model is now implemented in hardware, this leads to a discrepancy between the software modeling results and HIL results. The most noticeable difference is the presence of noise in the HIL simulation results and as such, non-consistent results might have to be filtered before further analysis and interpretation.

### 1. Sensitivity Analysis of Measure of Performance to Variations of Gain $K_1$ for Scenario A: Stationary Target

The sensitivity analysis of the measure of performance, $M_1$ to variations of gain $K_1$ is examined in a scenario where the target is stationary and the commanded range is 300m. The initial conditions are: (1) SUAV velocity = 28 m/s; (2) target velocity = 0m/s; (3) initial position of the SUAV is at [0, -1000, 300]; (4) initial position of the target is at [0, 0, 0] and (5) $K_2 = 0.20$. Figure 44 shows the variation of $\eta$, $\rho^*$, $\varepsilon$ errors and $M_1$ criterion with variations of gain $K_1$ for Scenario A while Figure 45 shows the error dynamics for the optimal case of $K_1 = 0.300$.

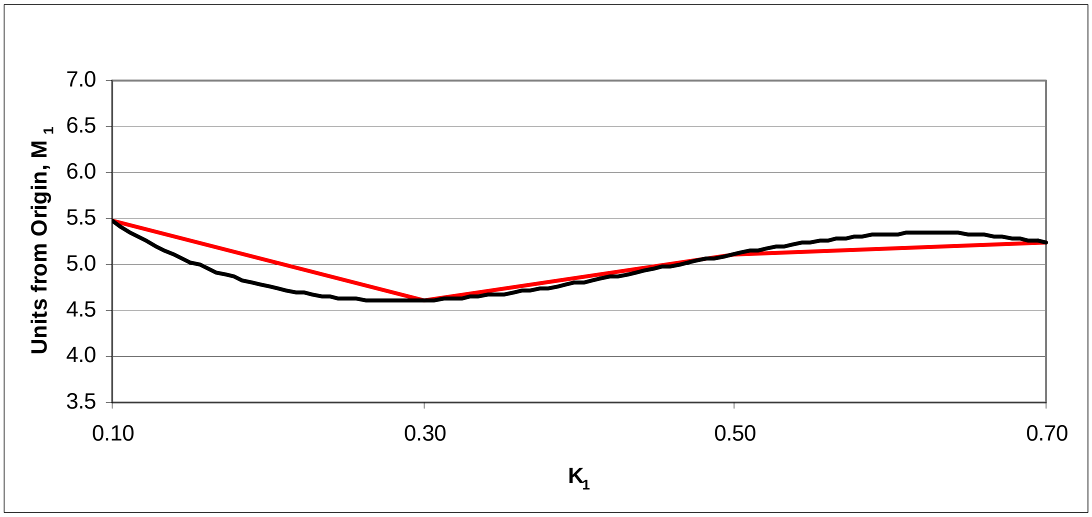
The following observations can be made from the plots:

(a) From Figure 44, it is observed that $M_1$ can also be minimized at a value of $K_1 = 0.30$. Compared to the SIMULINK results, the errors obtained in the HIL simulations are larger in value.

(b) From Figure 45, exceptionally high values (spikes) are observed in the $\eta$ and $\varepsilon$ error plots. These spikes occur several times but with each occurrence lasting for very short periods of time (almost instantaneously) before returning to the normal range of values. This is inherent for the HIL simulations due to the integration between software and several hardware components and should be filtered off in actual flight test.

Scenario A: Stationary Target (HIL)

Measure of Performance, $M_1$ with Variations in $K_1$




Performance of $\eta$, $\rho^*$, $\varepsilon$ and $M_1$ with Variations in $K_1$

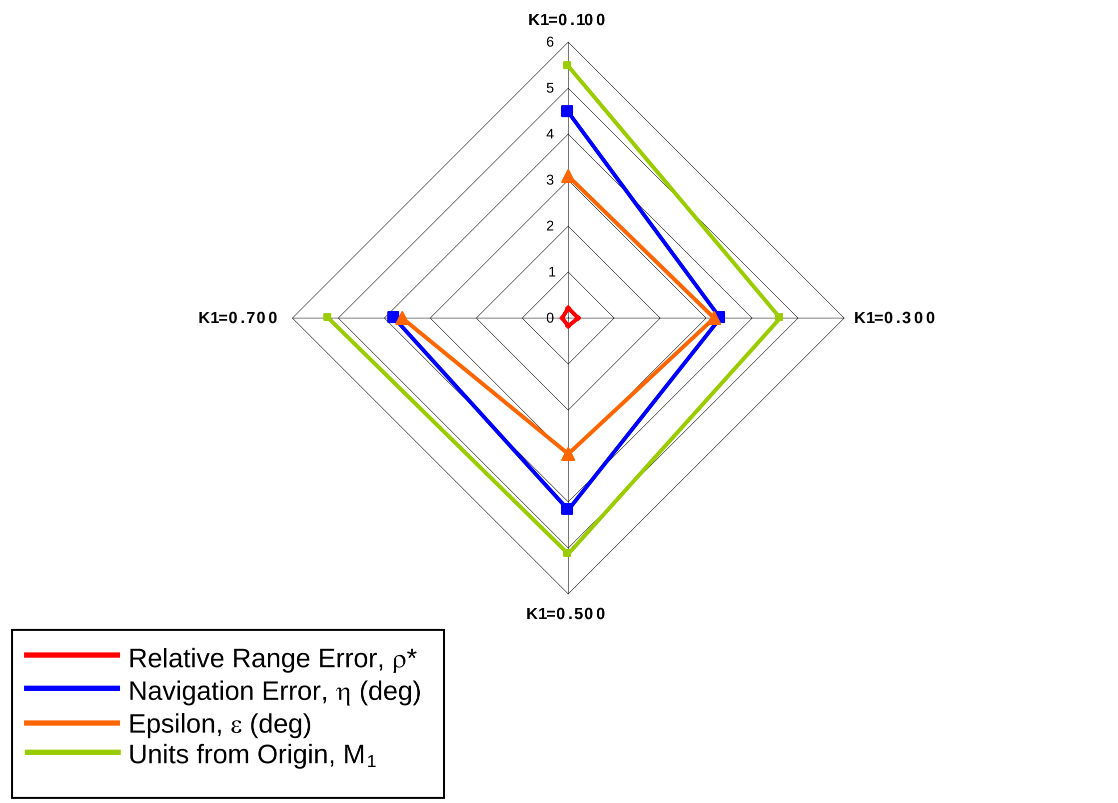
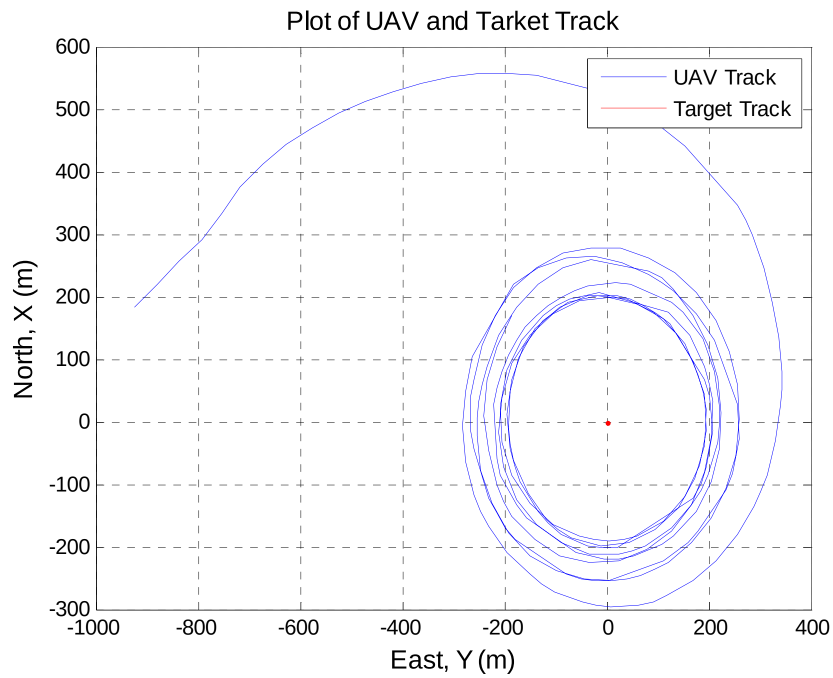


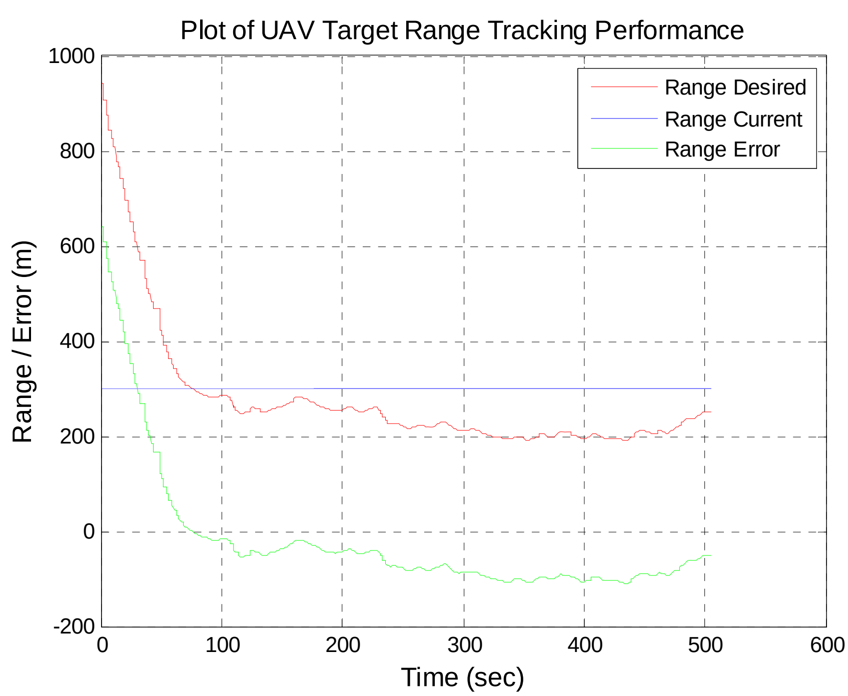

Figure 44. Variation of $M_1$ vs $K_1$ for Scenario A (HIL)

Scenario A: $K_1$ = 0.300 (HIL, Optimal Case)

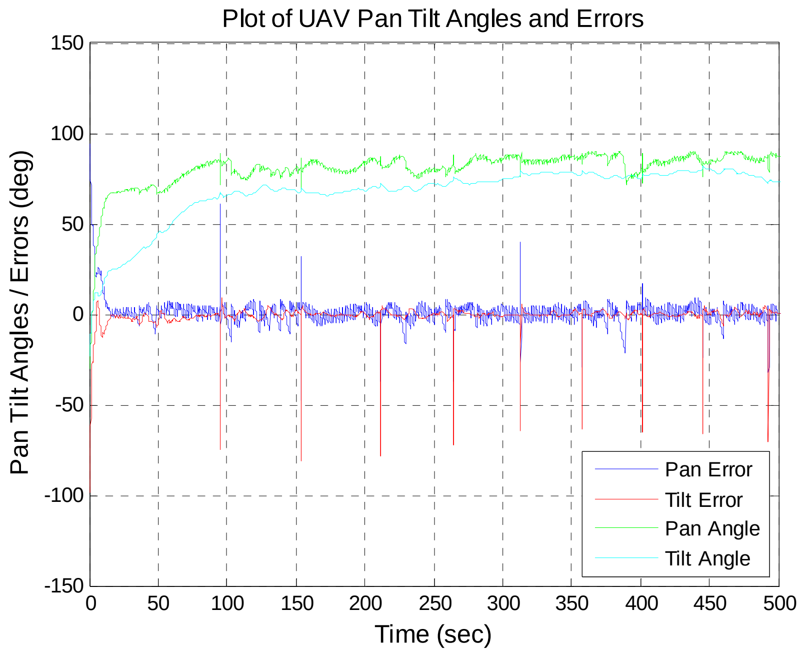
SUAV target VBTT control law tracking history




Range convergence history of the SUAV to the desired range of 300m



Pan / Tilt angles convergence history and gimbal camera angle errors




Navigation angle error convergence history

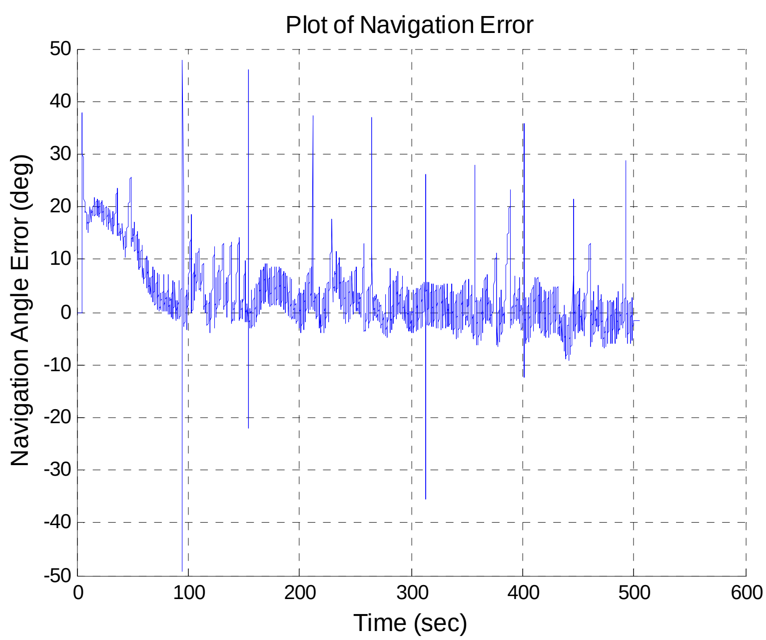



Figure 45. Sensitivity Analysis for $K_1$ = 0.300 for Scenario A (HIL)

### 2. Sensitivity Analysis of Measure of Performance to Variations of $K_1$ for Scenario C: Target is Moving with a Nominal Speed (8m/s) with Variable Heading

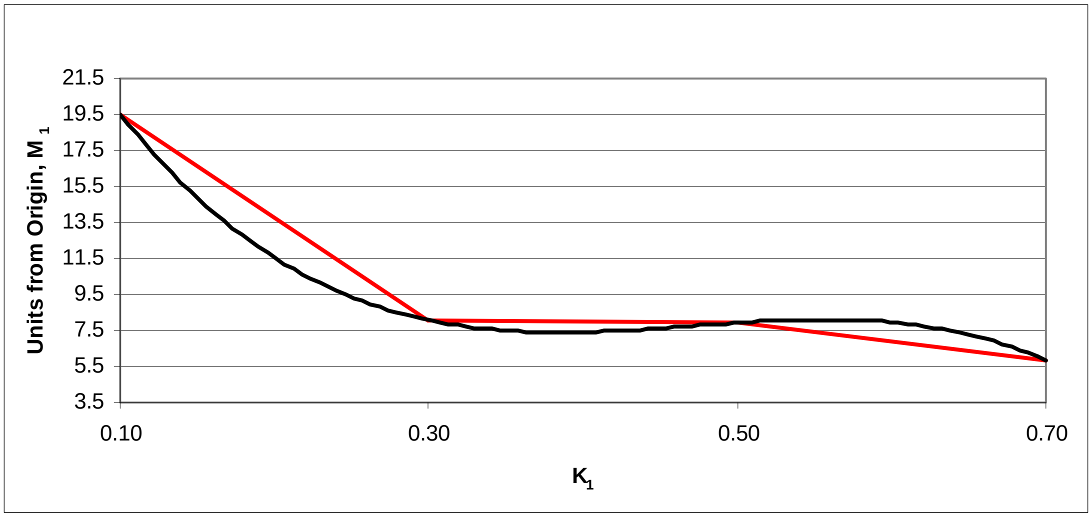
The sensitivity analysis of the measure of performance, $M_1$ to variations of parameter $K_1$ is examined in a scenario where the target is moving with a constant heading and the commanded range is 300m. The initial conditions are: (1) SUAV velocity = 28 m/s; (2) target velocity = 8m/s; (3) initial position of the SUAV is at [0, -1000, 300]; (4) initial position of the target is at [0, 0, 0] and (5) $K_2$ = 0.20. Figure 46 shows the variation of $\eta$, $\rho^*$, $\varepsilon$ errors and $M_1$ criterion with variations in $K_1$ for Scenario B while Figure 47 shows the error dynamics for the optimal case of $K_1 = 0.70$ for Scenario C.

The following observation can be made from the plots:

(a) From Figure 46, it is observed that value of $M_1$ decreases with larger values of $K_1$ and is minimized at a value of $K_1 = 0.70$.

Scenario C: Moving Target with Variable Target Heading (HIL)

Measure of Performance, $M_1$ with Variations in $K_1$




Performance of $\eta$, $\rho^*$, $\varepsilon$ and $M_1$ with Variations in $K_1$

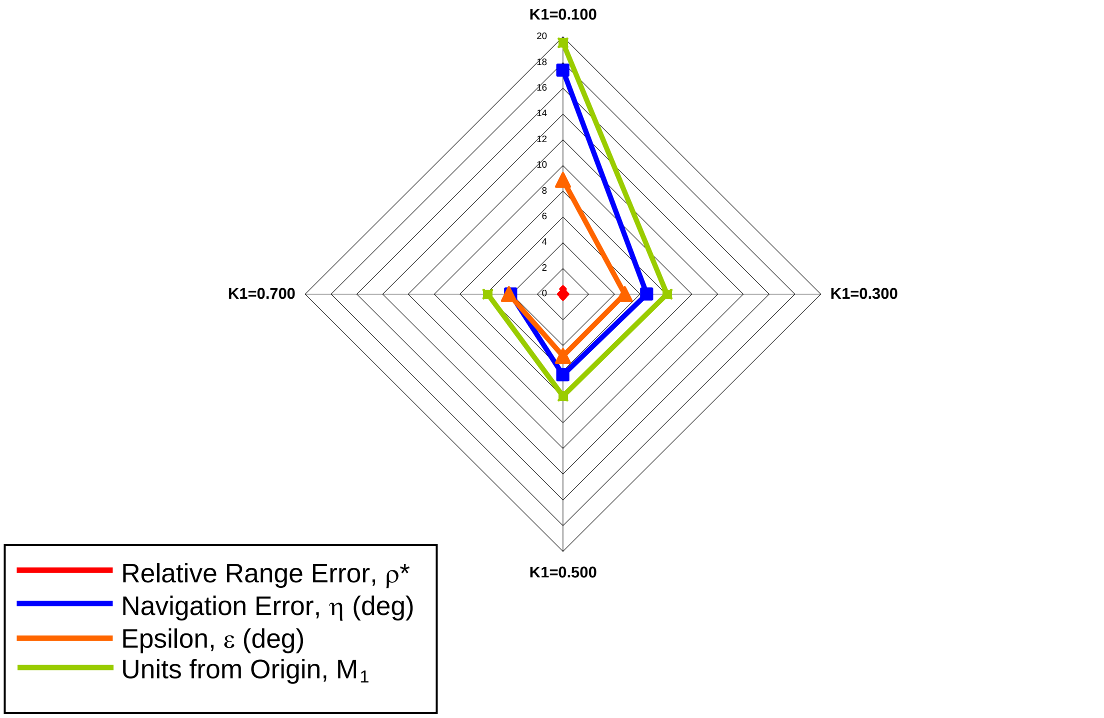
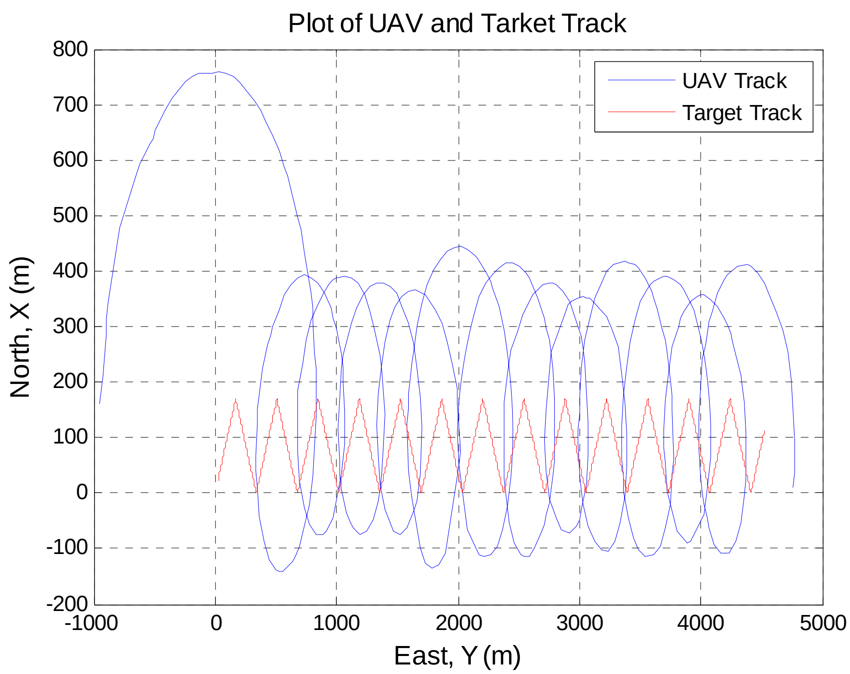


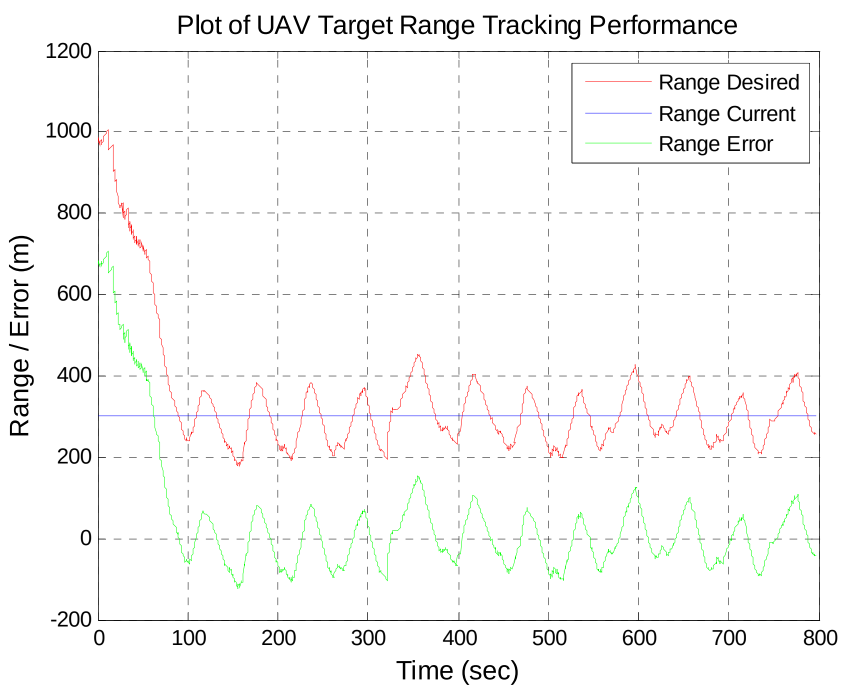

Figure 46. Variation of $M_1$ vs $K_1$ for Scenario C (HIL)

Scenario C: $K_1$ = 0.700 (HIL, Optimal Case)

SUAV target VBTT control law tracking history




Range convergence history of the SUAV to the desired range of 300m



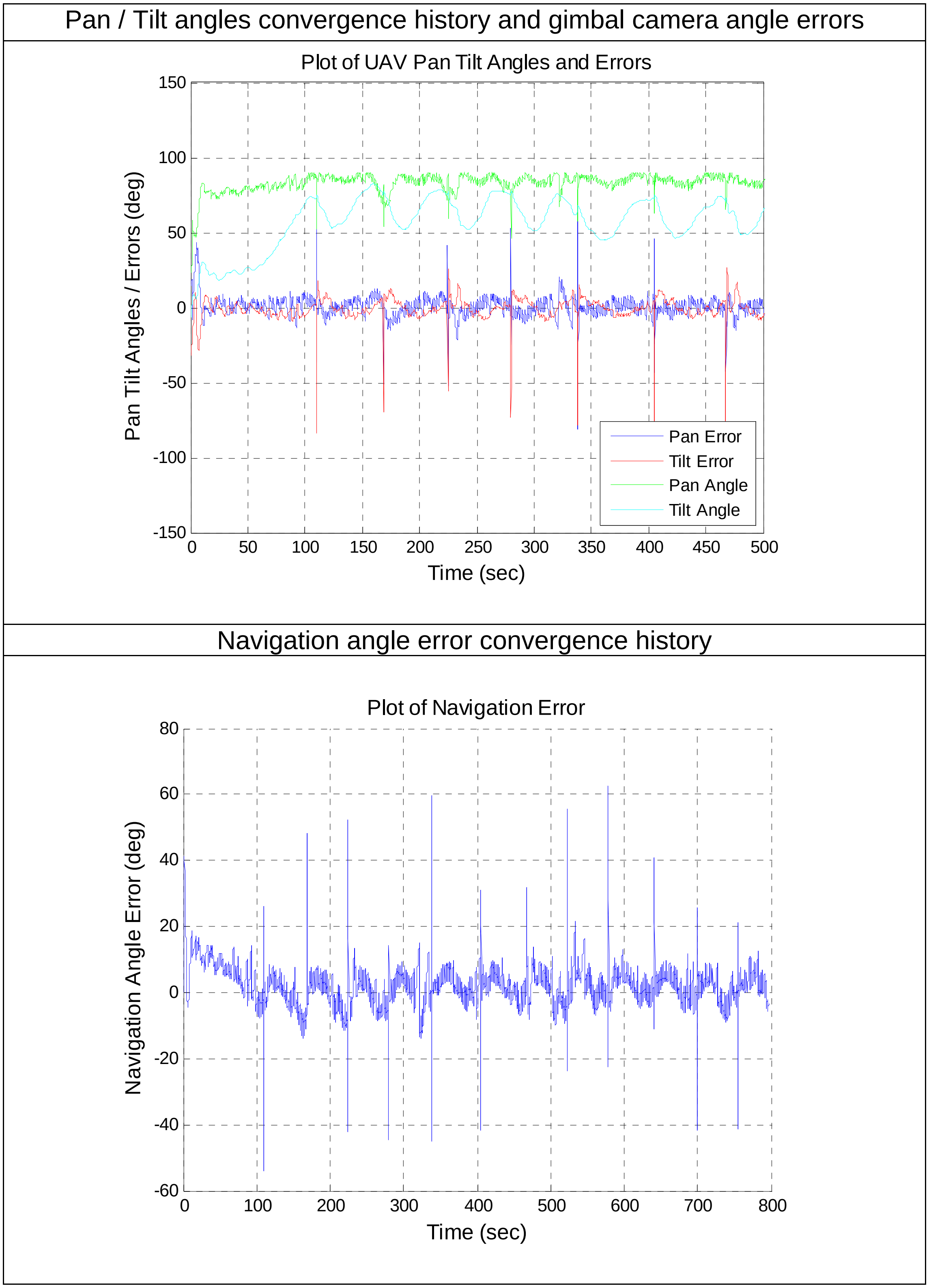



Figure 47. Sensitivity Analysis for $K_1$ = 0.700 for Scenario C (HIL)

### 3. Summary of HIL Simulation Results

#### a. *Variations of $M_1$ for Scenarios A and C*

Figure 48 summarizes the trends (indicated by the dotted lines) of $M_1$ with variations of $K_1$ for the two target maneuver scenarios in a HIL simulation environment. This is compared to the earlier results obtained from the SIMULINK numerical simulations as indicated by the solid lines. It is observed that the HIL results are in unison with the SIMULINK results. The key difference is the larger values of $M_1$ criterion for the HIL results, indicating higher values of $\eta$, $\rho^*$ and $\varepsilon$ errors. This is expected due to a noticeable level of noise present in the HIL simulations.

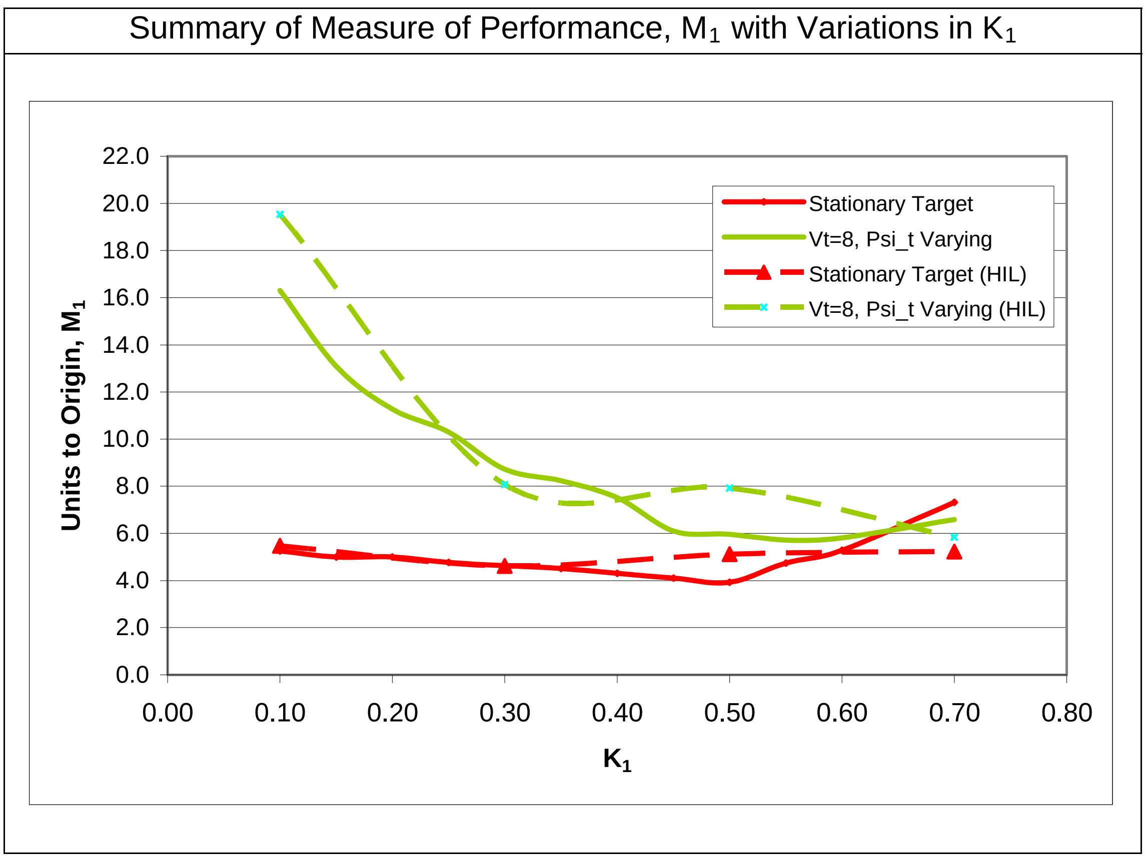



Figure 48. Plot of $M_1$ vs $K_1$ for Scenarios A and C

#### b. *Variations of $M_1$ for Scenario A1: Target is Stationary with Different Values of Time Delay*

Figure 49 illustrates the impact on $M_1$ with time delay. As indicated by the dotted lines, it is observed that the introduction of a time delay does not adversely affect the values of $M_1$ to the same degree as shown in SIMULINK-based results. This illustrates that the chosen software-based environment and results obtained are more conservative in nature; in turn the HIL results incorporating a real AP unit are more promising and representative of real flight implementation.

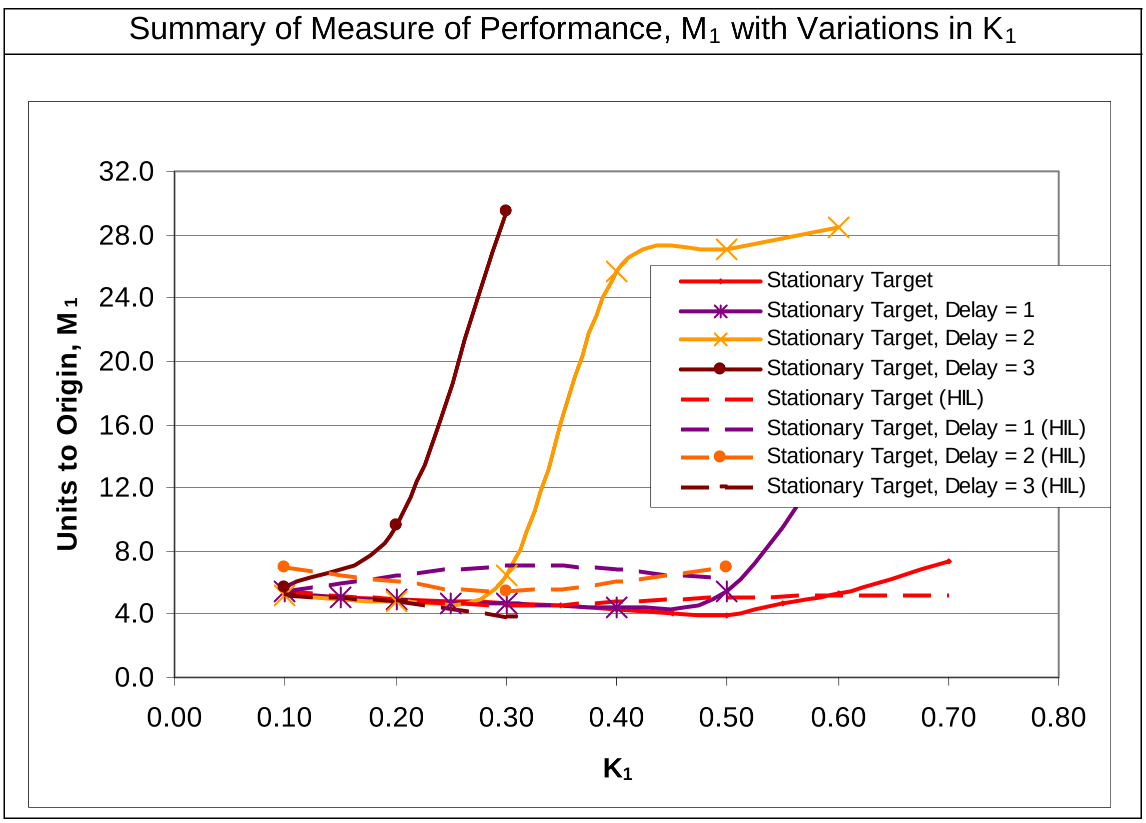



Figure 49. Plot of $M_1$ vs $K_1$ for Scenario A1 (HIL)

### c. *Variations of $M_1$ for Scenario C1: Target is Moving with a Nominal Speed (8m/s) with Variable Heading with Different Values of Time Delay*

Figure 50 illustrates the impact on $M_1$ with time delay to the system in a HIL simulation environment. As before the results are less conservative and therefore promise greater feasibility of flight implementation.

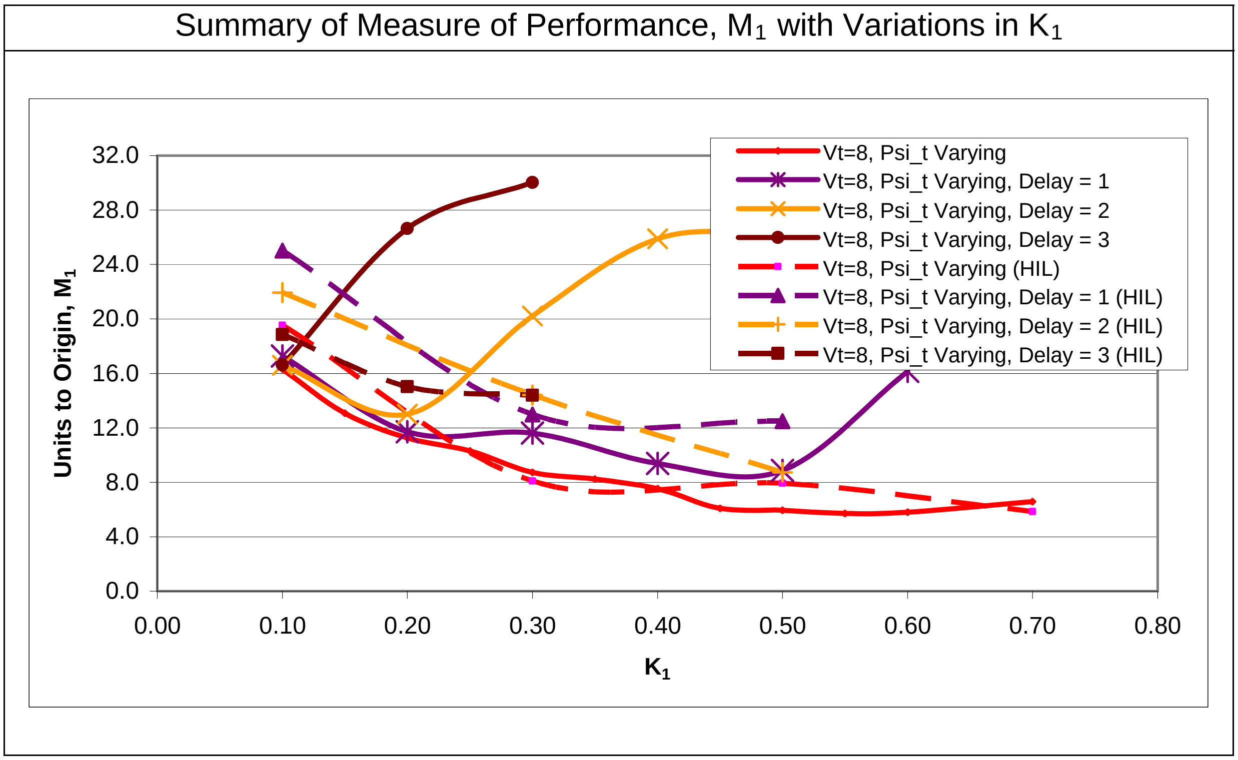



Figure 50. Plot of $M_1$ vs $K_1$ for Scenario C1 (HIL)

## E. HIL SIMULATION CONCLUSIONS

Although the obtained HIL results are noisier in general when compared to software simulation results, the HIL simulation results still exhibit similar performance for the stationary target (Scenario A) and target maneuver case (Scenario C). The differences in the HIL and SIMULINK results are due to the inherent differences in the AP and 6-DOF UAV models employed in the HIL simulation. The additional hardware noises that are present in the HIL simulation could also have attributed to the variations in the results. However, both the SIMULINK and HIL results have validated the performance of the control law for these two scenarios.

When time delays were introduced in Scenarios A1 and C1, the results obtained between SIMULINK simulations and HIL simulations do not exhibit the same performance and trend. This observation indicates that testing in software and hardware environments can be vastly different for certain scenarios. As such, it is essential to perform simulations in the HIL environment before any actual flight as HIL simulations better emulate actual flight dynamics better.

THIS PAGE INTENTIONALLY LEFT BLANK

# V. CONCLUSIONS AND RECOMMENDATIONS

## A. CONCLUSIONS

In this thesis, the robustness of the existing Vision Based Target Tracking (VBTT) adaptive guidance law was further analyzed in software simulations by considering scenarios where the target velocity, both magnitude and direction, were varied. Scenarios with time delay in feedback were also subsequently analyzed. The results indicated that there was no single value of the feedback gain $K_1$ that was able to satisfy all possible variations of target maneuver. This implied that the existing control law has not been designed to meet different operational scenarios and therefore it is necessary to develop an improved control law.

The results obtained in the hardware-in-the-loop (HIL) simulation environment generally exhibit the same performance as those obtained in software simulations for the non-time delay target maneuver scenarios. However, larger error values were obtained due to the presence of noise in HIL simulations and differences of the 6-DOF modeling of UAV dynamics. In addition, it should be noted that software and HIL simulation results were different when time delay was introduced. HIL results in the time delay scenarios were less restrictive in the values of the feedback gain and therefore promise greater feasibility in actual flight implementation.

## B. RECOMMENDATIONS

To complete the vision-based hardware-in-the-loop simulation setup in a laboratory environment, both the gimbaled TASE camera and the Pan-Tilt Unit hardware should be integrated. This will greatly improve the testing of future vision-based algorithms.

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF REFERENCES


[1] Vladimir Dobrokhodov, Issac Kaminer & Kevin D Jones and Reza Ghabcheloo (2006). Vision Based Tracking and Motion Estimation For Moving Targets Using Small UAVs. *American Institute of Aeronautics and Astronautics*. Retrieved August 2008.

[2] G. M. Siouris, Aerospace Avionics Systems*: A Modern Synthesis*. New York: Academic Press, 1993.

[3] B. Etkin and L. D. Reid. *Dynamics of Flight, 3rd ed*. New York: John Wiley and Sons, 1995.

[4] R. A. Prince, "Autonomous Visual Tracking of Stationary Targets Using Small Unmanned Aerial Vehicles." M.S. thesis, Naval Postgraduate School, Monterey, CA, U.S.A., 2004.

[5] T. C. Boon, "Development And Implementation of New Control Law For Visual Based Target Tracking System Onboard Small Unmanned Aerial Vehicles." M.S. thesis, Naval Postgraduate School, Monterey, CA, U.S.A., 2006.

[6] L. Ma, C. Cao, N. Hovakimyan, V. Dobrokhodov and I. Kaminer, "Adaptive Vision-Based Guidance Law with Guaranteed Performance Bounds for Tracking a Ground Target with Time-Varying Velocity" presented at AIAA Guidance, Navigation, and Control Conference and Exhibit, Honolulu, HI, 2008.

[7] K. Ogata, *Modern Control Engineering, 4th ed*. New Jersey: Prentice Hall, 2002.

[8] Laboratory for Intelligent Mechanical Systems (LIMS), "Introduction to PC/104 setup, building xPC model for Target PC," *Northwestern University*, 2006. [Online]. Available: http://lims.mech.northwestern.edu/students/stephens/me333/main.html. [Accessed October 2008].

[9] Computer Controlled Systems, "PC104 Embedded System FAQ," *Arrick Publishing*, 2006. [Online]. Available: http://www.controlled.com/pc104faq/. [Accessed October 2008].

[10] Directed Perception, "Computer Controlled Pan –Tilt Unit Model PTU-D300 User's Manual," Version 2.18.0, 2006. [Online]. Available: http://www.DPerception.com/. [Accessed October 2008].

[11] B. Vagilienti and R. Hoag, “Piccolo System User Guide,” *Cloud Cap Technologies*, Version 1.3.2., 2006. [Online]. Available: http://www.cloudcaptech.com/. [Accessed Dec 2008].

[12] Cloud Cap Technologies, “Piccolo Quick Setup Guide,” Cloud Cap Technologies, 2002. [Online]. Available: http://www.cloudcaptech.com/. [Accessed Dec 2008].

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Dr. Issac Kaminer
   Naval Postgraduate School
   Monterey, California

4. Dr. Vladimir Dobrokhodov
   Naval Postgraduate School
   Monterey, California

5. Dr. Kevin Jones
   Naval Postgraduate School
   Monterey, California

6. Dr. Knox Millsaps
   Chairman, Department of Mechanical and Astronautical Engineering
   Naval Postgraduate School
   Monterey, California

7. Professor Yeo Tat Soon, Director
   Temasek Defence Systems Institute
   National University of Singapore
   Singapore

8. Tan Lai Poh (Ms), Assistant Manager
   Temasek Defence Systems Institute
   National University of Singapore
   Singapore